Organ Segmentation Of Pediatric Computed Tomography (CT) With Generative Adversarial Networks

Chi Nok Enoch Kan
Marquette University

ORGAN SEGMENTATION OF PEDIATRIC COMPUTED TOMOGRAPHY (CT) WITH GENERATIVE ADVERSARIAL NETWORKS

by

Chi Nok Enoch Kan

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

December 2020

ABSTRACT
ORGAN SEGMENTATION OF PEDIATRIC COMPUTED TOMOGRAPHY (CT) WITH GENERATIVE ADVERSARIAL NETWORKS

Chi Nok Enoch Kan

Marquette University, 2020

Accurately segmenting organs in abdominal computed tomography (CT) is crucial for many clinical applications such as organ-specific dose estimation. With the recent emergence of deep learning techniques for computer vision, many powerful frameworks are proposed for organ segmentation in abdominal CT images. A major problem with these state-of-the-art methods is that they depend on large amounts of training data to achieve high segmentation accuracy. Pediatric abdominal CTs are particularly hard to obtain since these children are much more sensitive to ionizing radiation than adults. It is extremely challenging to train automatic segmentation algorithms on pediatric CT volumes. To address these issues, we propose 2 new GAN architectures for abdominal CT synthesis and a combined segmentation-synthesis network with a built-in auxiliary classifier generative adversarial network (ACGAN) that conditionally generates additional features during training. All 3 frameworks are tested on a pediatric abdominal CT dataset collected by the Medical College of Wisconsin. Both of our proposed GAN architectures can generate quantitatively and qualitatively realistic abdominal CT images and patches. 2.5D segmentation experiments with 4-fold cross validation confirms our proposed segmentation framework, CFG-SegNet, is indeed high-performing and able precisely segment reproductive organs in abdominal CTs across multiple patient ages.

ACKNOWLEDGEMENTS

Chi Nok Enoch Kan

I would like to extend my deepest gratitude to Dr. Dong Hye Ye for providing the valuable opportunity for me to complete my master's degree under his supervision. Without his guidance and constant encouragement this thesis research would not have been possible. I would like to thank my fellow students in the MLIP lab, in particular Najib Akram and David Helmaniak for helping me out with both hardware issues and research questions.

I would like to thank my committee members, Dr. Richard Povinelli, Dr. Henry Medeiros and Dr. Taly Gilat-Schmidt for their contributions to medical computer vision, and their insightful feedback to tremendously improve the quality of this thesis. I would like to extend my gratitude to Dr. Mahjeed Hayat and Dr. Ayman El-Refaie for their help throughout my graduate school journey.

I am grateful for the constant encouragement from Dr. Romas Kazlauskas at the University of Minnesota- Twin Cities and Dr. Akshay Chaudhari at the Stanford Medical School. I am truly inspired by you and am able to stay motivated throughout the course of my graduate studies because of your kind and supportive comments.

I would like to thank my parents for their unfaltering love and support for the past 25 years. I would like to thank Keri Petski, John and Angie Weinrich, who blessed me with a life of joy when I am not working on my research. My appreciation also extends to Adrian Buntrock, who is an incredibly brave and kind individual currently battling a high-grade glioblastoma. Finally, I am indebted to the Wong family (Kelvin, Michelle, Alfred, Esther Wong) for treating me as one of their own and for loving me for the past 7 years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Thesis Statement

Though advanced deep learning frameworks have been proposed for multi-organ segmentation of abdominal computed tomography (CT) images, class imbalance remains a prevalent problem in many clinical datasets. Segmentation performances are poor in pediatric datasets containing reproductive organs due to a significant lack of training images. This thesis proposes three novel frameworks based on generative adversarial networks (GANs) to synthesize new abdominal CT, segment reproductive organs, and thereby improve the current segmentation performance on pediatric abdominal CT datasets.

1.2 Common Medical Imaging Techniques

1.2.1 History of Medical Imaging

Medical imaging refers to a set of techniques used to create visual representations of the human body to aid clinical diagnoses. Popular medical imaging techniques include Magnetic Resonance Imaging (MRI), Computed Tomography (CT), X-Ray and Positron Emission Tomography (PET). On the other hand, electrical activity recording techniques such as electroencephalography (EEG) and electrocardiography (EKG) are considered medical imaging techniques as well since recordings can be expressed as feature maps [1]. However, these techniques are only considered as medical imaging in the most general sense and will not be discussed in this thesis.

The first concept of medical imaging came from the invention of X-Rays. German mechanical engineer Wilhelm Röntgen discovered the ability to “see through” his own

hand through an electron beam tube [2]. As he experimented and propagated the idea of using radiation to create unseen images of the human body, physicians quickly began to manipulate this technology to examine skeletal structures and related traumas in World War I. Röntgen named the radiation "X" given its unknown nature at the time of discovery. This rudimentary imaging technique is now commonly known as X-Rays and is widely used in clinics and hospitals.

1.2.2 X-Ray

X-ray is one of the most common medical imaging techniques used to locate fractured bones in the body or other lesions in soft tissues. Other applications of x-ray include but are not limited to mammography and dental examinations [4]. Patients undergoing a x-ray radiography are exposed to low-dose ionizing radiation and static images are generated from the exposure. As x-ray passes through the human body, they are absorbed at different rates in different parts of the human body. Therefore, a detector is placed on the other side of the human body to measure the flux and spectrum of the differentiated x-rays to produce images.

1.2.3 Computed Tomography (CT)

Computed Tomography (CT) is an extended application of X-Ray. The name "computed tomography" hints that it is essentially a computerized version of x-ray imaging. During the typical process of a CT procedure, a patient is exposed to x-ray beams that are rotated around the body. The beams that pass through the body are collected by rotating detectors and are subsequently computed to produce a cross-sectional slice of the patient. As the patient is moved along the rotating beams by a motorized table, an image volume can be constructed by stacking individual tomographic

images together. CT can be classified according to the dose of radiation: ultra-low-dose CT (ULDCT), low-dose CT (LDCT) and standard-dose CT (SDCT).

1.3 Deep Learning's Applications in Medical Imaging

1.3.1 Deep Convolutional Neural Networks

Deep learning is a subfield of machine learning which uses artificial neural networks (ANNs) to perform various representation learning tasks. Inspired by the human brain, ANNs are composed of interconnected nodes called neurons. The output of each neuron is computed as a linear function with weights and biases as its parameters. Non-linear activation functions such as sigmoid or hyperbolic tangent (tanh) functions are then used to map the linear output to a non-linear space. ANNs used in deep learning often have multiple layers and are considered as "deep" networks. Image recognition is a well-studied area of deep learning and deep convolutional neural networks (dCNNs) are developed for the purpose of object detection and segmentation in images. dCNNs make use of a simple mathematical operation known as the convolution operation to perform feature extraction. Fig. 1.1 shows a simple example of convolutional operation in a dCNN. Recall a given image can be expressed as a matrix of m x n pixels, where m is the number of rows and n is the number of columns. We can then use a kernel, which is nothing but a simple filter matrix to extract features from the input image. During the feature extraction process, the kernel slides across an input image and performs element-wise multiplication at each stride. The spatial size of the resulting feature map

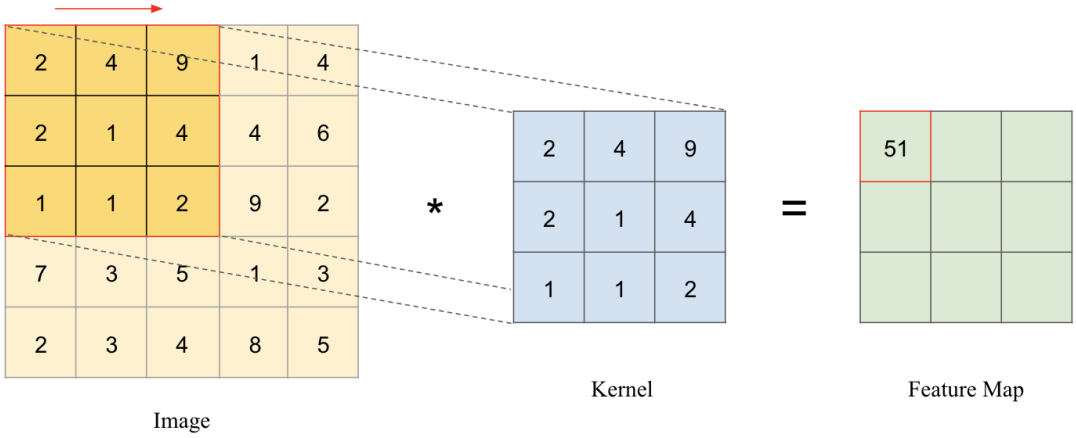


Figure 1.1 Simple convolution operation in deep convolutional neural networks. A kernel is used to convolve an image to produce a feature map. The '*' symbol represents matrix multiplication

can be calculated using Equation 1.1, where I is the input dimensions, O is the output dimensions, K is the size of the kernel, S is the stride and P is the padding. Note that padding is optional and can be used to create more space for the kernel to cover the entire image.

$$O = \frac{I - K + 2P}{S} + 1 \quad (1.1)$$

Using the same example illustrated in Fig. 1.1, we can computed the output dimensions by substituting I = 5 (5 x 5 input image), K = 3 (3 x 3 kernel), P = 0 (no padding) and S = 1 (single stride) into Equation 1.1 to get an output dimension of 3 x 3.

1.3.2 Object Detection and Segmentation in Medical Imaging

Before the advent of machine learning, detection and segmentation in medical imaging was often performed manually or using SURF (Speeded Up Robust Features) descriptors. SURF is an effective computer vision algorithm that is used to perform real-time object tracking, disparity computation, object detection, motion-based segmentation and 3D reconstruction using image registrations. SURF detectors aim to address

problems related to the SIFT (Scale Invariant Feature Transform) algorithm [5], which uses 128-dimensional descriptors and can potentially be computationally intensive. Making use of the fact that Hessian-based detectors are much more stable than their Harris-based counterparts, SURF uses a Hessian blob detector to find points of interest in each image.

As mentioned in the previous section, images can be represented as pixel intensity matrices. With the rising popularity of convolutional neural networks proposed by Yann LeCunn [6], researchers began to develop improved frameworks for medical image analysis. It goes without saying that object detection is an easier task than segmentation in medical imaging, as segmentation requires a given algorithm to accurately detect the boundaries of a region of interest (ROI). Ultimately, both object detection and segmentation are used for computer-aided diagnosis (CAD) in real-life settings to highlight lesions or other ROIs in medical images.

Region proposal networks (RPNs) are important backbone networks in object detection. RPN uses the concept of anchors, which are center points of a sliding window which travels across a given image to identify proposed regions. A typical RPN is a lightweight network made up of a single classifier and a single regressor, where the classifier calculates the probabilities that an image region contains the target object and the regressor calculates the coordinates of the proposed regions. A parameter k is used to determine the number of proposed regions for a given image, and the number of anchors is a multiple of the image dimensions (height*width) and k. R-CNN is a popular object detection framework which incorporates RPN to extract proposals, and subsequently classify features for each proposal's feature map using convolutional layers. A major

problem with R-CNN, later identified by researchers, is that R-CNN takes a huge amount of time to train since each proposed region is classified independently.

Faster R-CNN is proposed by the R-CNN's original creators to address the complexity issues in R-CNN. Instead of computing region proposals directly from input images, feature maps are extracted and used for ROI extraction. Region proposals are also transformed using a special ROI pooling layer. Fig. 1.2 shows an overview of the Faster R-CNN framework. Faster R-CNN has been applied to identify pathological features in chest x-rays [7] and detecting intervertebral discs in lateral lumbar x-rays [8].

The goal of image segmentation is to not only localize ROIs but to create pixel-wise masks of them. Most segmentation tasks related to medical imaging are semantic segmentation, as opposed to instance segmentation where multiple objects of the same class are treated as individual objects. Common network architectures used to perform semantic segmentation on medical images include: Fully Convolutional Network (FCN) and its variants, Mask R-CNN and U-Net. U-Net and other related encoder-decoder architectures have gained notoriety in recent years due to their incredible performances on biomedical image segmentation. Proposed by Ronneberger et al. [10], U-Net is an extension of FCN where feature maps in the encoder network are concatenated to their counterparts in the decoder network. As graphical processing units (GPUs) gain popularity in the field of deep learning due to their high computing power, the U-Net architecture has been extended to 3D U-Nets or V-Nets to perform volumetric segmentation [11]. Dense V-Net, a powerful variant of V-Net which uses dense feature stacks and strided convolutions are used to perform multi-organ segmentation in abdominal CT volumes [12].

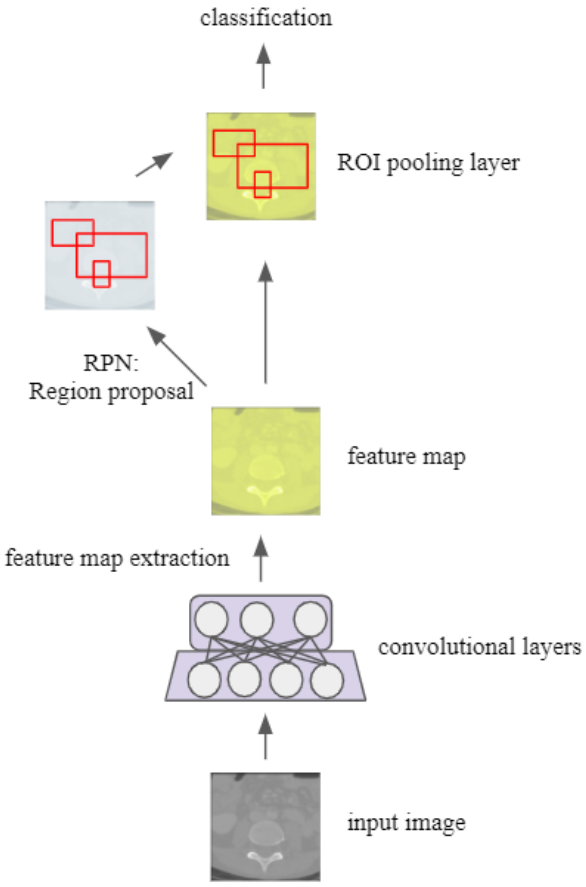


Figure 1.2 Faster R-CNN framework for object detection. Multiple region proposals are generated by RPN, and subsequently pooled via an ROI pooling layer.

1.3.3 Image Denoising and Super-Resolution

The simplest solution to image denoising is the use of non-linear filters, which suppress noise while preserving edges. In recent years, however, researchers are constantly working on new dCNN architectures for denoising in the field of medical imaging. A common problem in training dCNNs for medical image denoising is that there is a lack of noisy medical images available in real life. One way to tackle this problem is to use additive white noisy images (AWNIs), which includes multiple ways to distort training images such as the addition of Gaussian and Poisson noise. Usually, a single end-to-end CNN framework is used for image denoising and in some cases prior

knowledge is used [13]. Besides dCNNs, autoencoders have also been applied to denoise mammograms and dental x-rays [14]. With the recent breakthroughs in generative models, generative adversarial networks (GANs) such as fidelity-embedded GAN (f-GAN) are used to denoise unpaired low dose CT (LDCT) and standard dose CT (SDCT) images [15].

Another interesting application of dCNN on medical image analysis is image super-resolution. Before dCNNs were popular, interpolation techniques such as bicubic interpolation were commonly used to increase image resolution by enhancing spatial resolution. However, interpolation is quite limited and is extremely susceptible to noise in images. Hence, other super-resolution methods such as patched-based super-resolution and super-resolution reconstruction (SRR) are proposed to tackle these issues [16]. Upon discovering the effectiveness of dCNNs in capturing and learning image features, researchers have quickly found multiple use cases for GANs to enhance MRI resolutions [17]. Besides GANs, frameworks combining dCNNs and scaling algorithms have also been proposed to brain MRI images [18].

1.3.4 Image Synthesis

Medical image synthesis and modality transfer are new trends in medical image analysis. With the invention of GAN, researchers can synthesize high quality, viable and realistic images. Besides image synthesis from noise vectors, GANs are also capable of performing modality transfer. Two well-known conditional GAN architectures are Pix2Pix [19] and CycleGAN. They are often used to perform image-to-image translation tasks such as CT denoising and MR reconstruction. It is important to note that although these tasks are ultimately used to denoise and reconstruct images, their mechanisms rely

heavily on modality transfer (e.g. from noisy to denoised) [21]. Pix2pix architectures have also been modified to generate retinal images from vessel trees [22]. As for unconditional synthesis, deep convolutional GANs (DCGANs) and Wasserstein GANs (WGANs) are often used to generate organ lesions and segmentation masks. With NVIDIA's latest development on progressive growing of GANs (PGANs), researchers can synthesize high quality skin lesion images given semantic and instance maps [23].

1.4 Overview of Thesis Structure

All the work discussed and presented in this thesis is based on the fact that age imbalance is a common and unsolved problem in medical image segmentation. Accurately segmenting organs in abdominal computed tomography (CT) scans is crucial for clinical applications such as dose estimation for pre-operative planning. While many successful deep learning methods have been proposed for multi-organ segmentation in abdominal CT images, most of them require large amounts of training data to achieve high segmentation accuracy of up to 80 to 90%. Children are more vulnerable to ionizing radiation exposure in CT scans than adults, and hence very few pediatric abdominal CT images are clinically available. Therefore, there is a strong need for a robust segmentation framework that is capable of auto-generating new training images while maintaining a high segmentation accuracy for pediatric abdominal CT scans.

The arrangement of the thesis is as follows:

- Chapter 1 provides a summary of our work, and an introduction to various medical imaging modalities and current deep learning techniques.
- Chapter 2 reviews literatures on the current state of abdominal CT organ segmentation and its clinical usage.

- Chapter 3 details the methodologies and implementations of the two generative adversarial networks (BS-DCGAN, Age-ACGAN) proposed in this thesis.
- Chapter 4 details the methodologies and implementations of a single feature-generating segmentation framework (CFG-SegNet) proposed in this thesis.
- Chapter 5 reports experimental results of the proposed models on a private pediatric abdominal CT dataset.
- Chapter 6 summarizes this thesis and discusses potential extensions to our work.

This thesis proposes two modified GAN architectures for abdominal CT synthesis and a novel segmentation framework that effectively combines image synthesis and organ segmentation in abdominal CTs. Validation with a pediatric abdominal CT dataset containing 120 patients shows that our proposed models are high-performing, and our proposed CFG-SegNet is capable of segmenting difficult organs such as reproductive organs.

CHAPTER 2
BACKGROUND

2.1 Clinical Significance of Abdominal CT Organ Segmentation

2.1.1 Dose Estimation

As mentioned in section 1.2.3, radiation dose in CT is extremely important and has always been a primary research interest in the medical field. Accurately computing organ dose is vital to ensure the safety of patients. Rapid dose quantification can be done by combining dose maps calculated from Monte Carlo-based simulations and organ segmentation masks generated from a U-Net [25]. Fig. 2.1 shows a brief outline of the proposed organ dose estimation method:

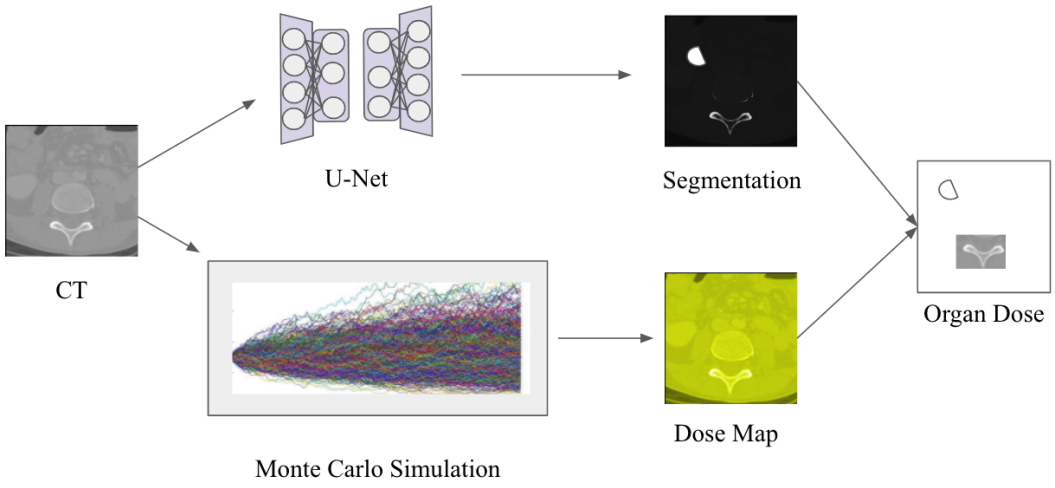


Figure 2.1 Organ dose computation framework. Monte Carlo simulation is used to produce dose maps which can be coupled with organ segmentation to produce organ-specific dose maps.

Another similar approach by Taly et. al. [25] uses Smart Segmentation Knowledge Based Contouring, an auto-segmentation algorithm coupled with dose maps computed with the

Monte Carlo method to generate patient-specific organ dose estimates. The auto-segmentation algorithm effectively combines atlas-based and feature-based segmentation methods. Another example of using an auto-segmentation algorithm to compute organ dose is the use of a volumetric CNN with 3D convolutional layers [26]. The computed left and right kidney segmentation masks are coupled with volumetric dose maps, which are previously computed from single photon emission computed tomography (SPECT) scans. This framework provides an effective way to estimate renal radiation dose in abdominal CT scans.

2.1.2 Preoperative Planning

Preoperative planning, also known as surgical planning, is the process of visualizing surgical interventions prior to a surgery. Preoperative planning is important for surgeries that are more invasive or have higher risks. In orthopedic surgeries, CT segmentation is particularly important during the preoperative planning phase as it provides vital navigation information to the surgeons. For surgeries that aim to remove tumors from vital organs in the abdomen, CT segmentation is often used during preoperative planning to identify the position of the tumor [27]. Segmentation can also be used during surgical simulation of the preoperative planning phase. For example, a surgeon can simulate a resection plane on the hepatic lobe of a patient with metastases from colorectal cancer.

2.2 Current State of Abdominal CT Organ Segmentation

2.2.1 Pixel-wise Segmentation

Pixel-wise segmentation is the segmentation of 2D images. Pixel-wise organ segmentation of CT scans is well-studied, and many high-performing architectures have

been proposed in the past. A method frequently used by clinicians to produce segmentation masks for a given CT scan is atlas-based segmentation. A common goal of atlas-based segmentation methods is to use prior knowledge from one or more atlases to produce new segmentation masks. To understand how atlas-based segmentation works, one must understand the concept of image registration.

Image registration is simply the transformation of multiple sets of images from various angles into a consolidated coordinate plane. A commonly used image registration algorithm is linear transformation, where two images are aligned based on a set of affine transformations such as rotation and shifting. Similarly, if we are given an existing CT image from patient A with a manually labelled liver segmentation mask, we can then create an image registration between patient A's CT image and an unseen CT image from patient B. Using the same transformation algorithm we used to transform the CT images, we can then transform the existing liver segmentation mask to produce patient B's liver mask. In practice, multiple atlases from different patients are used to enrich prior knowledge since anatomical variabilities are high in certain organs.

U-Net has gained tremendous popularity for its ability to accurately segment biomedical images. U-Net follows an encoder-decoder architecture, and different networks can be used for its encoder and its decoder. Fig. 2.2 shows the general architecture of U-Net:

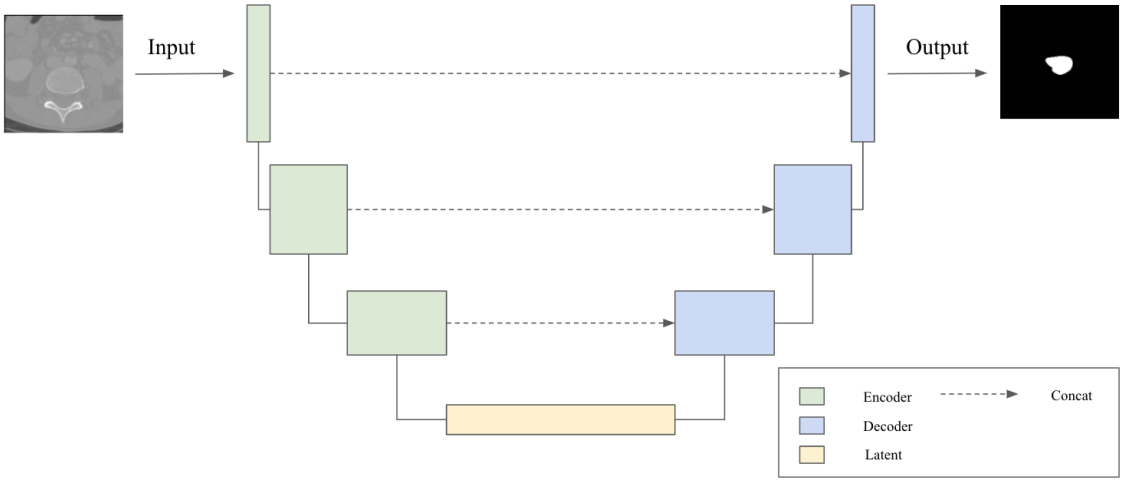


Figure 2.2 U-Net architecture. U-Net architecture is considered an encoder-decoder architecture which relies on multiple skip connections

The encoder part of U-Net provides a contracting path that extracts the latent features in a given image (or voxel volume in 3D U-Net). The decoder part is an expansive path that transforms the latent features back to the original image dimensions through multiple convolutional layers. What is interesting about the U-Net architecture is that each layer in the encoder is concatenated to its corresponding layer in the decoder. These skip connections are commonly used in deep networks such as ResNet and make training very deep neural networks possible [28]. U-Net has inspired many other architectures for 2D CT organ segmentation, such as the dilated residual U-Net (DR U-Net) [29] and organ-attention networks with reverse connections (OAN-RCs) [30]. OAN-RC in particular has achieved state-of-the-art segmentation performances for 13 organs in abdominal CT scans.

Context-encoder network (CE-Net) is a new variant of U-Net which uses dense atrous convolution (DAC) blocks to preserve spatial information [31]. CE-Net can simultaneously take in images from multiple modalities and encode them through a pre-

trained feature encoder. Spatial information is preserved at the latent feature level using a DAC block and a residual multi-kernel pooling (RMP) block. Like the U-Net, the resulting features go through an expansion path in the decoder and segmentation masks for multiple modalities are produced. It is possible to perform multi-organ abdominal CT segmentation with CE-Net by treating organs as different modalities.

2.2.2 Volumetric Segmentation

Volumetric segmentation with deep networks is made possible by advancements in computing hardware. In the past, various techniques have been used to approximate volumetric segmentation masks for multiple organs. One such technique is the use of prediction-based segmentation [32]. Canonical correlation analysis is first used to find the spatial interrelations among abdominal organs, and a statistical atlas is later added explicitly to improve segmentation accuracy.

Another non-deep learning volumetric segmentation method is multi-atlas label fusion (MALF) [33]. MALF relies on label atlases based on image registration. Joint label fusion (JLF) is then used to reduce the correlation error in organ atlases. However, this method is hard to perform in practice since it requires image registration between multiple subjects.

With the invention of U-Net came the invention of V-Net, a volumetric variant of the popular biomedical image segmentation algorithm. Volumetric convolutions are used in V-Net to extract features from image volumes and are used in place of pooling operations to reduce memory usage [34]. Originally used to perform prostate segmentation for 50 MRI volumes, V-Net has been modified to segment other types of medical images. One such architecture is Dense V-Net, which is essentially a FCN with

dense feature stacks and convolutions. Fig. 2.3 shows the overall architecture of Dense V-Net.

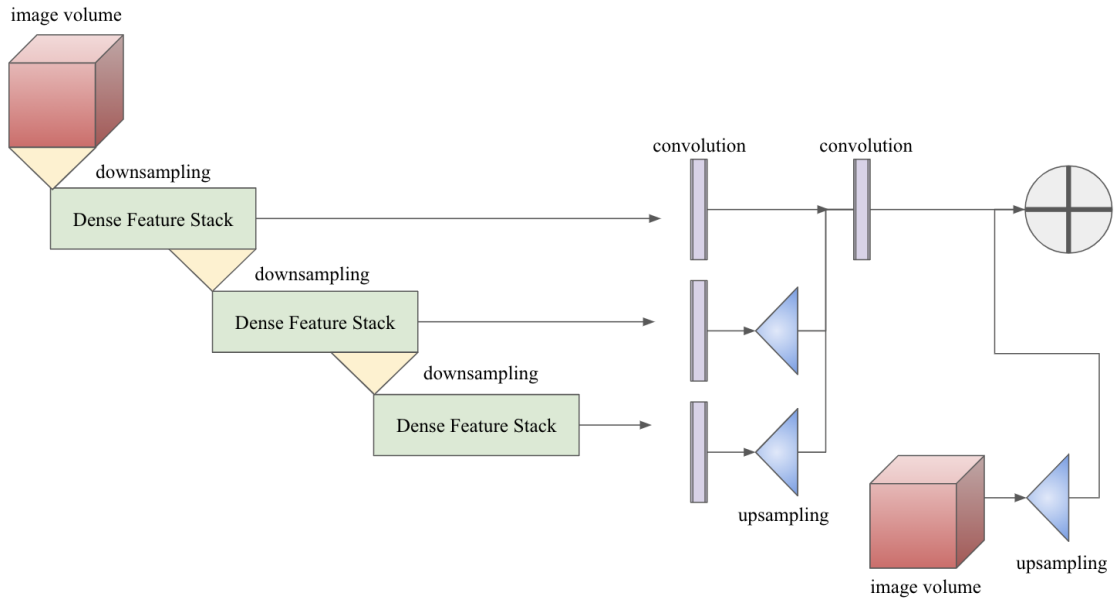


Figure 2.3 Dense V-Net architecture. The input image volume is downsampled via multiple dense feature stacks. Note that the original image volume is concatenated to the upsampled feature map

Dense V-Net has a unique architecture in which a cascade of dense feature stacks is used to generate three separate activation maps. These activation maps are then normalized to the same resolution and concatenated with a spatial prior, which is generated by upsampling the original image volume.

Segmentation performance on abdominal organs depends on the availability of training images and other factors such as variability in shapes and sizes. For example, it is much harder to obtain a good segmentation performance for the uterus because the uterus has a huge anatomical variability. Most of the dCNN-based segmentation algorithms also require large amounts of data to train in order to capture the target data distributions well. Pediatric CT volumes are in practice quite hard to obtain, and hence

segmentation performance on pediatric organs has also been quite poor. Table 2.1 summarizes the general applications of a few segmentation algorithms on different abdominal organs:

Table 2.1 Organ segmentation applications of deep learning-based algorithms. Only two of the reported experiments include reproductive organ segmentation

Method	Esoph.	Gall.	Kidney	Liver	Panc.	Prost.	Spleen	Uterus
2D-3D-FCN [35]	✓	✓	✓	✓	×	×	✓	×
3D-CNN [36]	×	×	✓	✓	×	×	✓	×
Regional ConvNet [37]	✓	✓	✓	✓	×	×	✓	×
3D-FCN [38, 39]	✓	✓	✓	✓	×	×	✓	✓
VoxRes Net [40]	✓	✓	✓	✓	×	×	✓	×
3D-Patch-based-dCNN [41]	×	×	✓	✓	×	×	×	×
Global-Local-CNN [42]	×	×	×	×	×	✓	×	×
Dense V-Net	✓	✓	✓	✓	✓	×	✓	×

2.2.3 Limitations

As mentioned in the end of the previous section, current segmentation algorithms are limited by two major factors: availability of medical images and anatomical variability of the organ. It is generally much harder to find images containing organs at risk that are more radiosensitive than other organs, such as the prostate and the uterus. In fact, only one of the algorithms reported in Table 2.1 have performed segmentation on the uterus. Segmentation performance on these hard-to-find organs are also highly dependent on the quality of the annotated mask by physicians. This makes a great use case for GANs or self-supervised dCNNs to work with organs that are generally underrepresented in training images.

As technology rapidly progresses, hardware limitation is becoming less of a problem. However, it remains a fact that medical images are often high-definition and consume a huge amount of disk space. Generalizability and practicality of the algorithms are also questionable as most hospital computer systems are quite archaic, and it will certainly take some time before deep learning algorithms are completely streamlined into CAD workflows in the field of medicine. As imaging devices and scanners become more advanced, medical images also increase significantly in their resolutions. This creates the need for more computing power, as algorithms such as U-Net tend to have an upper limit for their optimal resolutions.

2.3 Generative Adversarial Networks (GANs)

2.3.1 Data Augmentation with GANs

As mentioned in the previous section, class imbalance is a major issue when training deep learning algorithms to perform segmentation tasks on medical images. A

common technique to tackle class imbalance in training data is to geometrically augment existing images. Data augmentation is the use of geometric and other transformations to create new images from existing images. It has been a proven technique to improve the generalizability of the deep learning models. However, the optimal ways to augment training images are often unknown to researchers prior to training and hence extensive tuning is needed. Moreover, the number of ways to augment a given image is limited and it is impossible to create unlimited amounts of images simply using data augmentation.

This led to the invention of generative adversarial networks (GANs) [43]. GANs have gotten a lot of attention recently due to their ability to synthesize realistic images from white noise vectors. The originally proposed GAN architecture consists of two dCNNs competing against each other. Fig. 2.4 shows the architecture of a GAN in the most general sense:

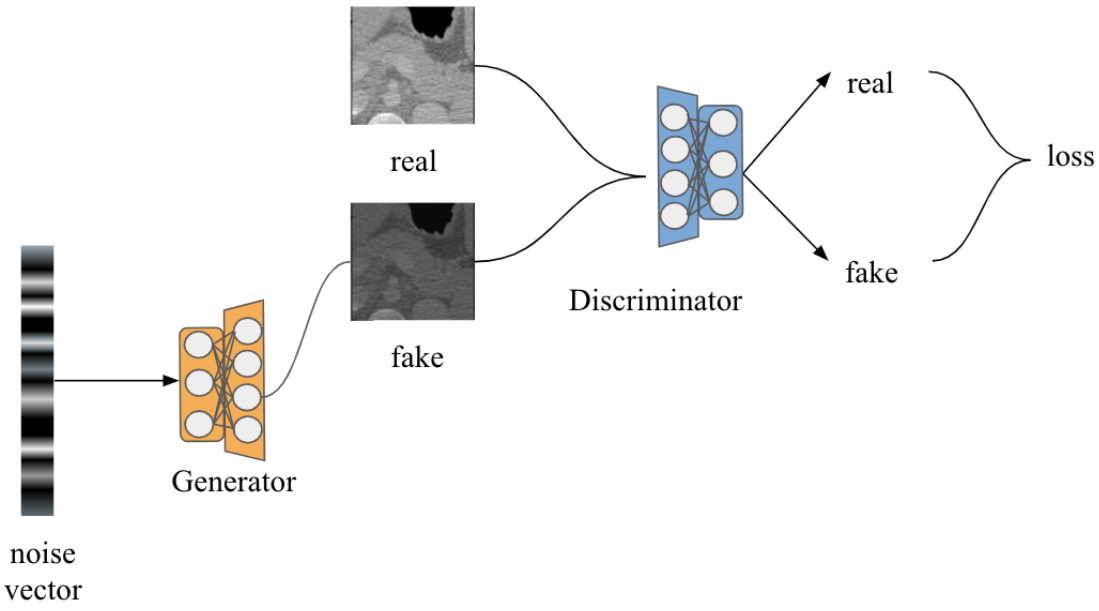


Figure 2.4 GAN architecture. A 'fake' image is generated by the generator from a random noise vector. The discriminator is tasked with classifying 'fake' images from real training data.

From Fig. 2.4, the two competing dCNNs are: a generator network that generates synthetic images from noise and a discriminator network that discriminates real samples from fake ones. A common learning objective of GANs is for the generator to 'fool' the discriminator with fake images that increasingly become realistic. Mathematically, we can express this learning objective as a minimax loss:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}\left[\log\left(1 - D\left(G(z)\right)\right)\right] \quad (2.1)$$

From Equation 2.1, the training objective of the discriminator (denoted as D) is to maximize log(D(x)) + log(1-D(G(z))), the probability of assigning correct labels to both training images and synthetic (or 'fake') images generated by the generator network G. The generator is trained to minimize log(1-D(G(z))), the inverted log probability of the discriminator's prediction of fake images. Minimization of the inverted probability is hard to implement and therefore we seek to maximize D(G(z)) instead.

The original GAN is capable of synthesizing realistic images. However, it can only synthesize them in a random fashion and is often susceptible to mode collapse. Mode collapse happens when the generator decides to pick the 'easiest' class in the dataset to successfully fool the discriminator. The resulting images lack diversity and usually they all belong to the same class. In practice, mode collapse happens very often due to class imbalance in training data. We will further discuss mode collapse in the following sections. One way to tackle mode collapse and to correct the problems appearing in unconditional image synthesis is to incorporate side information. Conditional GAN (cGAN) is a common type of GAN that uses a generator which conditionally generates images based on class labels [44]. Auxiliary classifier GAN (ACGAN) is a type of cGAN which uses an additional auxiliary classifier to assign the

correct class labels to synthesized images [45]. Fig. 2.5 shows the general architecture of ACGAN:

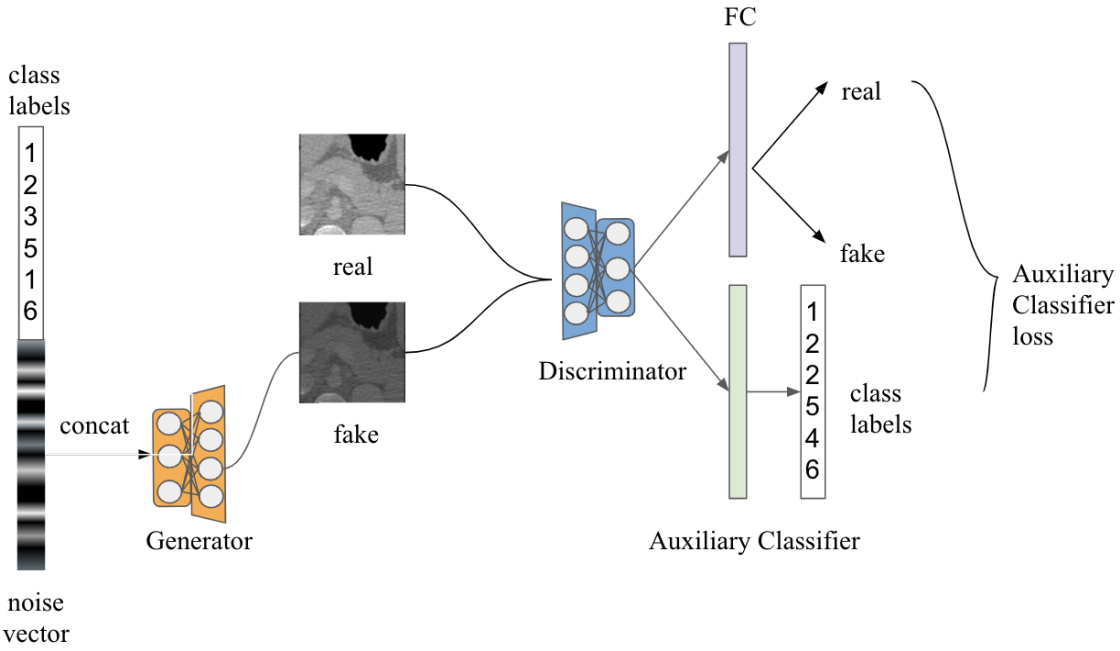


Figure 2.5 ACGAN architecture. ACGAN uses additional class labels during image synthesis to not only conditionally synthesize images, but to improve the quality of the synthesized images.

Since ACGAN includes an additional auxiliary classifier in its discriminator, its objection functions are defined as the follows:

$$L_s = E[\log P(S = \text{real}|X_{\text{real}})] + E[\log P(S = \text{fake}|X_{\text{fake}})] \quad (2.2)$$

$$L_c = E[\log P(C = c|X_{\text{real}})] + E[\log P(C = c|X_{\text{fake}})] \quad (2.3)$$

Besides producing a probability distribution P(S|X) = D(X) over possible images sources, ACGAN's discriminator also contains an auxiliary classifier that produces a probability distribution P(C|X) = D(X) over the class labels of the images. The objective functions 2.2 and 2.3 are defined as the log-likelihood of the correct image source L_S and the log-likelihood of the correct class L_C.

Pix2pix is a popular variant of cGAN which is commonly used in image-to-image translation tasks. It is built upon the well-known segmentation network U-Net and uses adversarial learning to achieve modality transfer. In pix2pix, the generator is usually a U-Net (or any other encoder-decoder networks) and the discriminator is a convolutional "PatchGAN" classifier. Unlike other cGANs, pix2pix uses a dual objective function which combines adversarial loss with L1 loss:

$$G^* = \text{argmin}_G \text{max}_D L_{cGAN}(G, D) + \lambda L1 \quad (2.4)$$

The first term L_{cGAN} represents the loss function of cGAN. This term can be substituted with any other cGAN loss functions, but Equation 2.1 is normally used. The second term, L1(G) represents the pixel-wise reconstruction loss measured by L1 loss (Mean Absolute Error) [46]. λ is simply a tunable parameter which changes the ratio between the two loss functions. When λ is set to be 0, the loss function G^* becomes cGAN's loss function.

It goes without saying that progressive growing of GANs (PGAN) is one of the most successful high-resolution image synthesis frameworks at the moment. Originally developed by Nvidia Research, PGAN is a scalable GAN architecture that is capable of synthesizing high-resolution images by progressively growing the resolution of both Generator and Discriminator layers [47]. Nvidia's publication in ICLR 2018 indicates that PGAN is capable of synthesizing high-definition facial images up to 1024 × 1024. In general, PGAN's generator aims to produce images starting from a lower resolution such as 4 × 4. Upon reaching convergence, the generator then scales up to a higher resolution with wider and deeper layers. PGAN scales up for each resolution by projecting network layers to higher resolutions using nearest neighbor interpolation. It also uses minibatch standard deviation to prevent mode collapse and equalized learning rates to stabilize

network training. One major drawback of PGAN, however, is its training time. Training PGAN on CelebA-HQ takes almost up to 2 days using 8 GPUs. Moreover, conditional synthesis is not available for PGAN and hence its generated images are sampled randomly. Fig. 2.6 shows the general idea of image synthesis with PGANs:

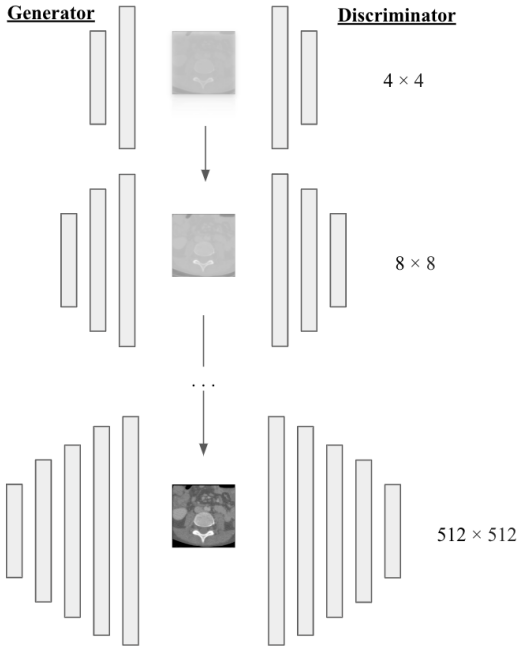


Figure 2.6 PGAN architecture. Starting from the lowest possible resolution, GAN layers are progressively grown to increase the resolution of the synthesized images

As technology advances and better computational hardware is available, more complex GAN architectures such as StarGAN and Dual generator GAN (G^2GAN) where multiple generators and discriminators are present become available [48]. It is impossible to summarize all the available GANs in this thesis, but Table 2.2 provides a brief summary of some of the most commonly used GANs in research:

Table 2.2 Common GAN architectures and their uses. This list is not exhaustive and only includes the most common GANs used in medical research

Method	Description

DCGAN[49]	• Unconditional image synthesis with deep convolutional layers
Pix2Pix	• Image-to-image translation
Cluster-GAN[50]	• Clustering in latent space
CycleGAN	• Image-to-image translation • Unpaired data
DualGAN[51]	• Image-to-image translation • Unsupervised dual learning
WGAN[52]	• Unconditional image synthesis • Wasserstein/ Earth-Mover distance metric
ACGAN	• Conditional image synthesis • Auxiliary classifier in discriminator
DRAGAN[53]	• Unconditional image synthesis • Gradient penalty scheme for training stabilization
StarGAN[54]	• Image-to-image translation • Multi-domain translation
DiscoGAN[55]	• Image-to-image translation • Unpaired data • Discovery of cross-domain relations

2.3.2 Existing Challenges in GAN Training

As mentioned in previous sections, one of the common challenges in GAN training is mode collapse. The problem of mode collapse has been well-studied and several methods including minibatch discrimination and guided latent spaces have been proposed to tackle it. In unconditional image synthesis with GAN, a variety of output is expected, and the synthesized images are expected to cover all of the original data distribution. However, if the generator decides to pick a specific class that is easiest to

learn it may ‘collapse’ and may only produce images from that class. Since the generator is focused on producing images from a single class, the quality of the produced images is extremely good but lacks diversity. On the other hand, the discriminator thinks that the generator is producing realistic images and the resulting losses from both networks will converge. Below are some examples from our experiments where mode collapse happens:

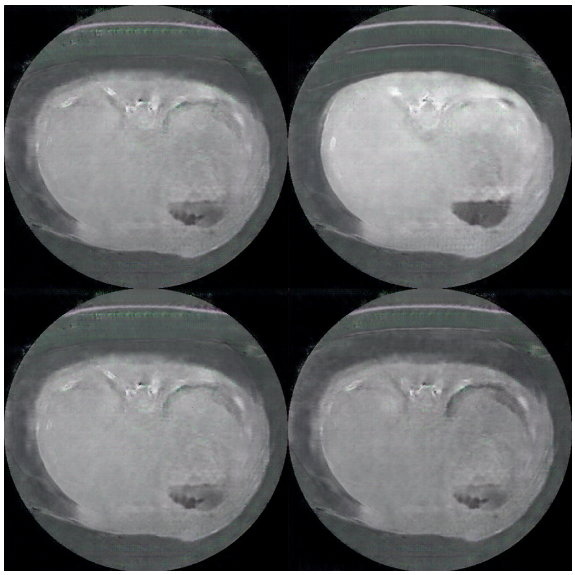


Figure 2.7 An example of mode collapse in GAN training. Notice the synthetic images lack variety.

A new loss function, called the Wasserstein loss is proposed in order to tackle mode collapse. Wasserstein loss uses a set of formulas derived from earth-mover distance and is defined as follow:

$$\max_{w \in W} \mathbb{E}_{x \sim \mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z \sim \mathbb{P}_z}[f_w(g_\theta(z))] \quad (2.5)$$

Unlike the original minimax loss defined in Equation 2.1, Wasserstein loss measures the earth-mover distance between real and fake data distributions. Its discriminator in fact does not classify whether a given image is real or fake, but rather outputs a value that is

maximized should the image be in fact real. It is interesting to mention that in Wasserstein GAN (WGAN) the discriminator is called a 'critic' instead of a discriminator since it does not perform classification. WGAN aims to solve mode collapse by training its critic to reject generator samples when a local minimum is not reached, thereby drastically reducing the possibility of mode collapse where the generator only produces one type of image.

Besides WGAN, another way to combat mode collapse is through the use of minibatch discrimination or minibatch standard deviation. The idea of minibatch discrimination [56] is simple: instead of discriminating between individual samples, we discriminate entire minibatches of samples. This simple change can massively dampen the effects of mode collapse, since the discriminator can easily detect low entropy (randomness) in generated images if it discriminates whole batches of samples. Minibatch standard deviation is usually used in PGANs, and it basically works the same way as minibatch discrimination does but instead calculates the standard deviation within each minibatch.

Another issue commonly encountered during GAN training is non-convergence. Little is known about non-convergence, but sometimes both the generator and discriminator networks simply fail to learn meaningful patterns in the training dataset and fail to converge. One of the more successful methods in getting GANs to converge is by adding instance noise sampled from a Gaussian distribution to discriminator inputs [57]. This technique aims to disrupt the lower bound in Jenshen-Shannon (JS) divergence during GAN-training, so the discriminator does not overfit. Besides the addition of instance noise, one-sided label smoothing is also used by the same authors where random

image labels are flipped during training. These methods aim to provide randomness to both the generator and the discriminator during training and ‘points’ them to the right direction to avoid non-convergence.

Gradient instability is also a major problem in GAN training. In short, gradient in deep learning is defined as a vector which gives the direction of steepest ascent/ descent of the loss function. The two possible cases when gradient is unstable are exploding gradient and vanishing gradient. When gradient explodes, derivatives are usually large and cause the gradient to accumulate throughout layers of the deep network. On the other hand, vanishing gradient happens when the calculated derivatives become infinitesimal. Both issues are related to poorly built models where each model update creates instability in the calculated loss. A common way to solve the problem of exploding gradients, proposed by the original creators of WGAN is to clip the gradients. On the other hand, using residual layers can help prevent vanishing gradients since skip connections allow gradients to be passed through multiple layers.

One challenging aspect of training GANs with volumetric layers, such as the 3D-GAN is the imbalance between the generator and the discriminator [58]. Since the discriminator is assigned a much easier task (real vs. fake classification) compared to the generator (volumetric synthesis), it tends to have an edge over the generator and eventually wins out after several epochs. The authors of the 3D-GAN paper propose a useful way to balance out the two networks. In their paper, they suggest only training the discriminator when its accuracy falls below a given threshold (usually somewhere between 70% to 80%). This allows the generator to update more frequently and to have sufficient opportunities to learn from the feedback given by the discriminator.

Finally, GANs are also extremely sensitive to their hyperparameters and often require extensive tuning to output ideal images. Learning rates between the generator and the discriminator often require balancing. Similar to other deep learning algorithms, there does not exist a 'one-size-fits-all' learning rate that works for any given dataset. However, one may find the use of Adam optimizers helpful since their adaptive momentum calculations eliminate the need for any learning rate scheduling and are hence less sensitive to their initial learning rates [59].

2.4 Challenges in Building Deep Learning Models Using Clinical Data

2.4.1 Scarcity of Pediatric Medical Images

Studies have shown that frequent visits to hospitals and clinics at a young age can lead to psychological trauma and emotional disorders in children [60]. Despite major advancements in modern medical technology, most medical imaging techniques remain quite invasive to children. Pediatric CT imaging is associated with radiation exposure risks and potential long-term damage such as the development of cancer later in life. It is a well-known fact that children are more susceptible to ionizing radiation than adults, and hence low doses of radiation are usually recommended for children. However, pediatric patients that require multiple scans may be exposed to unsafe amounts of radiation even at lower doses. Ultimately, it is important to understand that there is not an absolute dose which guarantees both the short-term and long-term safety of pediatric patients. A study done by the NIH (National Institutes of Health) have found that for a cumulative dose of anywhere between 50 to 60 mGy to the head, a 3-fold increase in brain tumor development risk is found in pediatric patients. The same study has also found that this critical cumulative dose level also increases the risk of developing leukemia by 3-fold.

Until better radiation-free imaging techniques are developed, clinicians will always consider the pediatric patient's long-term safety as their number one priority when performing imaging techniques. Currently, there are a few methods proposed to reduce the level of radiation exposure in pediatric patients and they are summarized in Table 2.3:

Table 2.3 Proposed methods to reduce radiation risks in pediatric patients [62]. Only X-Ray and CT are included in this table.

Method	Description
X-Ray: Beam Filtration	• Aluminum and copper filters in x-ray beams
X-Ray: Anti-Scatter Grid	• Produce less scattered radiation • Improvement in image quality
X-Ray: Strict Protocols	• Imaging staff must follow strict protocols to avoid errors leading to repeated radiation exposure
X-Ray: Radiation shields	• Breast shield • Thyroid shield • Gonadal shield • Fetal shield
CT: Reduction in Scanning Parameters	• Field of View (FoV) • Kilovolts • Rotation time • Detector coverage
CT: Adequate Insulation	• Blankets or warmers must be provided
Head CT: Perform Only When Necessary	• Head CT is only performed for trauma evaluation or as shunt protocol • All other neurological evaluations can be done using MRI

Abdominal CT: Perform Only When Necessary	• Abdominal CT imaging is only performed following trauma or abnormalities in organs • Spiral CT techniques can be used to evaluate abdominal lesions • If possible, ultrasound can be used in place of CT scans • Other MR techniques can also be used in place of abdominal CT

2.4.2 Logistical Difficulties in Implementation of Deep Learning Algorithms

There are quite a few logistical issues related to the implementation of deep learning algorithms in the medical field. Healthcare data is often not readily available for deep learning due to inconsistencies in data and formatting issues. For instance, it is hard to train and implement a segmentation algorithm if a hospital has a dataset where half the patients have missing DICOM (Digital Imaging and Communications in Medicine) headers. DICOM is the international medical imaging standard format which stores not only the images themselves but also the patients' information. Fig. 2.8 shows an example of a typical DICOM header:

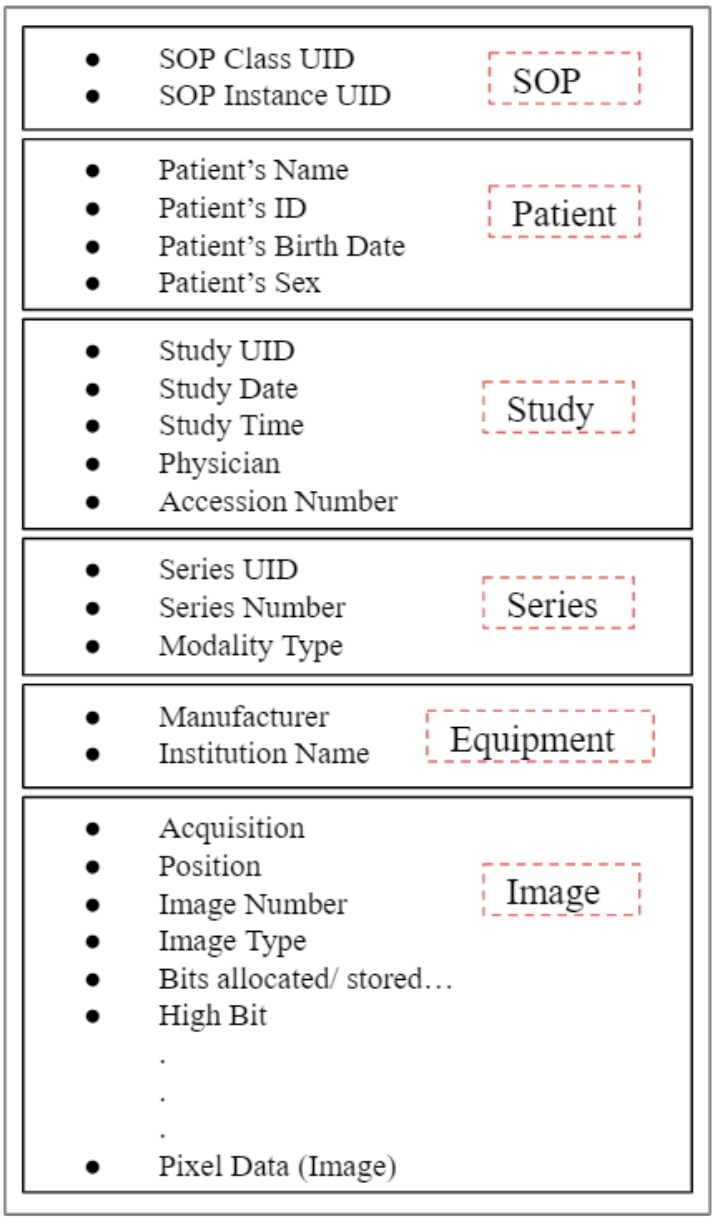


Figure 2.8 Example DICOM header. DICOM headers contain information about the patient and the details of the imaging study. Position, bit depth, pixel spacing, and other image data are included in the last section.

As shown above, there are many items other than the actual image that are encoded within the header of the image. Since deep learning models that are highly generalizable often depend on multiple predictors such as a patient's age or a patient's disease history,

missing headers in images present a huge problem to deploying these algorithms in clinical settings.

Although many of the hospitals in the world are equipped with advanced imaging machines, most of their computer storage systems remain archaic and outdated. It is hard to justify the purchase of high-end machines with GPU clusters to run deep learning algorithms. This major hurdle has prompted a shift in deep learning to focus on the development of lightweight and highly deployable algorithms with few parameters, such as MobileNet and EfficientNet [63]. EfficientNet is a class of dCNN architectures which uses mobile inverted bottleneck convolutional (MBConv) layers along with squeeze-and-excitation optimizations to achieve state-of-the-art classification accuracies. Moreover, the original creators of EfficientNet have shown its ability to scale vertically and in parallel without dramatically increasing the number of parameters. Ever since the invention of EfficientNet, its variants have been applied to solve various medical imaging problems such as diabetic retinopathy (DR) detection [64] and lung carcinoma classification [65].

CHAPTER 3
GENERATIVE ADVERSARIAL NETWORKS FOR ABDOMINAL CT SYNTHESIS

3.1 BatchNorm-SELU Deep Convolutional Generative Adversarial Network (BS-DCGAN)

3.1.1 Overview

Before we start constructing a robust segmentation framework with a built-in GAN, we need to have a stable GAN architecture which synthesizes high quality and viable CT images. Moreover, the GAN must be able to handle images with high CT pixel resolutions up to 512 × 512. A good candidate for this task is the deep convolutional generative adversarial network (DCGAN). As suggested by its name, DCGAN is a GAN variant which uses convolutional and deconvolutional layers instead of fully connected layers in its generator and its discriminator. DCGAN was originally proposed to resolve common issues related to the vanilla GAN architecture such as mode collapse and non-convergence. Unlike the original GAN, DCGAN has the following features:

- No fully connected layers in both the generator and the discriminator
- Batch Normalization in both the generator and the discriminator
- Fractional strided convolutions in generator
- Strided convolutions in discriminator
- Generator uses rectifier activation function
- Discriminator uses leaky rectifier activation function
- Generator uses either a tanh or a sigmoid as its output activation function

Note that the generator can use either a tanh or a sigmoid as its output function. This only affects the pixel intensity range in the synthesized image. However, it has been argued

that tanh provides a wider range of pixel intensity values for the generator's output and is therefore more ideal than sigmoid. Given all the beneficial aspects of DCGANs, we adapt a DCGAN with batch normalization and SELU (scaled exponential linear units) layers to perform unconditional abdominal CT synthesis.

3.1.2 Batch Normalization

The discovery of batch normalization is groundbreaking in deep learning research. Batch normalization is one of the four major normalization techniques, along with instance normalization, layer normalization and group normalization. Batch normalization enables deep networks to train without having gradient issues. In fact, studies have found that it speeds up the training process as well [66]. By shifting and scaling pre-activation layers, batch normalization essentially reduces the covariance shift in hidden units. The implementation of batch normalization layers in a deep network is quite simple. Before each activation layer, the output from the previous layer is divided into batches where each batch is subtracted by its own mean and divided by its standard deviation. Batch normalization also takes in two trainable parameters γ and β since it is undesirable to have normalized values in a narrow range between 0 and 1. Upon normalization, the normalized layer is shifted by β and scaled by γ. Since these two parameters are trainable, it is possible to train a network purely consisting of batch normalization layers to learn image representations. In fact, a recent study has found that training BatchNorm (batch normalization) layers alone in residual networks (ResNet101, ResNet110) can achieve classification accuracy of as high as 60% on benchmark datasets such as CIFAR-10 [68]. This is why DCGAN’s architecture incorporates BatchNorm layers to stabilize and accelerate training. In general, DCGAN uses convolution blocks

(the name convolution blocks can be confusing since it contains other non-convolutional layers), which contains a single convolution layer followed by a BatchNorm and an activation layer.

The mathematical basis of BatchNorm is simple and consists of four steps. Given the shift parameter β, the scaling parameter γ and a mini-batch B with n values, we first calculate the batch mean:

$$\mu_B = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{3.1}$$

Then we calculate the mini-batch variance:

$$\sigma_\beta^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_B)^2 \tag{3.2}$$

And normalize each input in the mini-batch B to get normalized inputs x_{norm}:

$$x_{norm} = \frac{x-\mu_B}{\sqrt{\sigma_\beta^2+\epsilon}} \tag{3.3}$$

Finally, we shift and scale with our two trainable parameters β and γ to get output y:

$$y = \gamma \cdot x_{norm} + \beta \tag{3.4}$$

Another important function of batch normalization is to regularize the network. Regularization is important when training deep networks since unregularized networks tend to overfit and hence fail to generalize on unseen data. Equation 2.8 creates a regularization effect by inducing sample noise through division and subtraction of batch mean and standard deviation.

3.1.3 Scaled Exponential Linear Unit (SELU)

Activation function is an important component of deep neural networks which computes the weighted sum of a given artificial neuron. As suggested by its name, activation functions determine whether artificial neurons should be fired or not. One

simple implementation of activation function is a binary threshold function which gives a value of either 0 (no activation) or 1 (activated). However, in practice we often require our activations to have more flexibility in how activated artificial neurons are in order to easily generalize and adapt to different tasks. A good solution is to use a non-absolute threshold function such as the rectifier function:

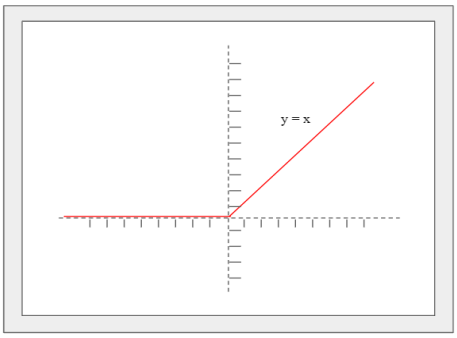


Figure 3.1 Rectifier activation function. All negative values are mapped to zero.

Mathematically, the output of a rectifier function is defined as the maximum of 0 and the input x, i.e. max(0, x). Compared to traditional activation functions such as the sigmoid function, rectifier functions are generally better in propagating gradients and providing sparse activation in networks. However, it does not guarantee absolute stability in gradients and can lead to a problem known as the 'dying ReLU' problem. A ReLU that is considered 'dead' will always output a constant value regardless of its input and is extremely unlikely to recover from this state. In comparison, leaky rectifiers give more flexibility in inactive units:

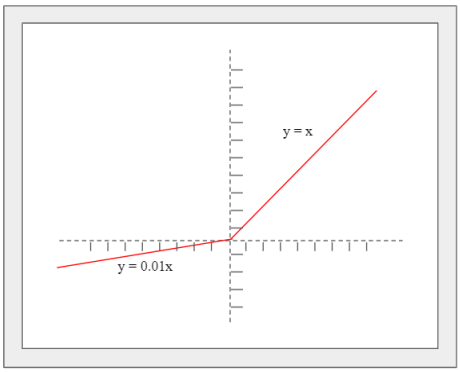


Figure 3.2 Leaky rectifier activation function. A small coefficient is used to 'leak' negative values

Leaky ReLU is controlled by a small slope α, which controls the magnitude of 'leakage' when the unit is inactive. This allows a small gradient to be leaked and thereby reducing the likelihood of dying linear units.

A new variant of ReLU called SELU (Scaled Exponential Linear Units) is proposed in self-normalizing networks (SNNs). A SNN is a type of FCN (fully convolutional network) which uses self-normalizing activation layers, where individual neuron activations are capable of normalizing themselves towards zero mean ($\mu = 0$) and unit variance ($\sigma^2 = 1$) via the following function:

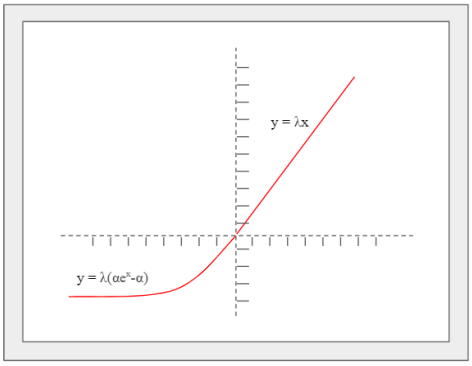


Figure 3.3 SELU activation function. A fixed parameter λ is used to scale positive and negative values.

As shown in Figure 3.3, SELU uses two fixed parameters λ and α. Unlike batch normalization, these two parameters are not trainable and hence no backpropagation happens through them. Similar to leaky RELU, SELU controls gradient by providing internal normalization when a unit is not active. When the units are active, i.e. when x is positive it is scaled by a fixed parameter λ. The idea of SELU is to normalize each activation layer without explicitly adding additional normalization layers such as instance normalization layers or batch normalization layers. In stark contrast with ReLUs, SELUs are capable of stabilizing gradients and do not have a 'dying' problem when gradients become extremely negative. In fact, SNN architectures that have achieved state-of-the-art results in the past only consist of a few layers since the use of SELUs has tremendously increased the networks' convergence speed and stability.

3.1.4 BatchNorm-SELU (BS) Layers

It is an unusual idea to combine SELU and BatchNorm layers since SELU is an internal normalization layer whereas BatchNorm is an explicit normalization layer. However, past research has shown that using SELU and BatchNorm together can lead to even faster convergence speeds and stability in training deep generative neural networks [68]. Theoretically, SELU is supposed to keep individual activation means $\mu_{activation}$ close to 0 and unit variances $\sigma^2_{activation}$ close to 1. However, even the most advanced modern GPUs are prone to floating-point arithmetic errors and their computational results are not entirely reproducible [69]. More specifically, GPUs that are running neural network simulations may introduce rounding errors since there are multiple threads running multiple synapses in different orders. Therefore, batch normalization provides an additional normalization layer to keep the means and variances close to [0,1] respectively

and thereby alleviating the effects of random noises. This proposed combination of SELU and BatchNorm is theoretically sound and is empirically determined to increase convergence speed more than when they are used separately.

3.1.5 Network Architecture

We adapt a deep convolutional generative adversarial network (DCGAN) with BS layers to generate high-definition medical images. The proposed BS-DCGAN generator architecture is shown below:

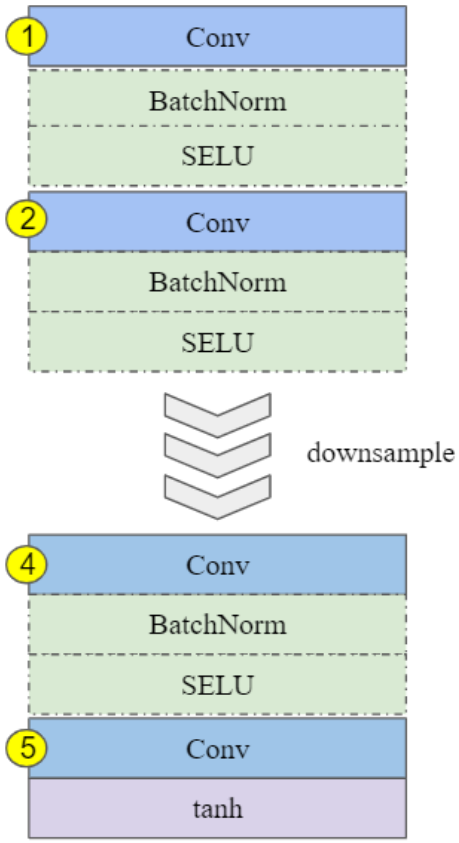


Figure 3.4 BS-DCGAN generator architecture. There are 4 downsampling BS-layers before the final convolutional layer.

No significant changes are made to the discriminator, and hence most discriminator layers are unchanged when compared to the original DCGAN except all leaky RELU

layers are replaced with SELU layers. Note that in our proposed BS-DCGAN generator architecture, an extra layer (layer 5) is added before the final tanh activation layer. We choose to use tanh instead of sigmoid for our output layer since it is capable of mapping a wider range of pixel intensity values. It is important to mention that our loss function is identical to that of both the vanilla and DCGAN's, which is formulated as a minimax loss defined in Equation 2.1.

3.2 Age Auxiliary Classifier GAN (Age-ACGAN)

3.2.1 Overview

A major drawback of BS-DCGAN is its inability to conditionally synthesize images. Since BS-DCGAN is an unconditional GAN, it synthesizes a variety of images sampled across the input data distribution. To conditionally synthesize novel yet viable pediatric abdominal CTs, we propose a new conditional GAN (cGAN) architecture which conditions on a patient's age. Specifically, we adapt an auxiliary classifier GAN (ACGAN) to accurately produce abdominal CT images of patients of young ages.

3.2.2 Training Objectives

The training objective of Age-ACGAN is different from that of BS-DCGAN's. Since ACGAN has an additional classifier attached to its discriminator, we have to take into account the classification accuracy of not only the real-fake images but their respective class labels as well [70]. In the original training objective of ACGAN, the discriminator not only produces a probability distribution $P(S|X) = D(X)$ over possible images sources but also produces a probability distribution $P(C|X) = D(X)$ over the class labels of the image. Recall from equations 2.2 and 2.3 that the overall objective function of ACGAN is defined as the log-likelihood of the correct source L_s and the log-

likelihood of the correct class L_C, we can now modify it as Age-ACGAN's loss function, which computes the log-likelihoods of the source L_S and the age class label L_A being assigned correctly:

$$L_S = E[\log(P(S_{CT} = \text{real}|X_{\text{real}}))] + E[\log(P(S_{CT} = \text{fake}|X_{\text{fake}}))] \quad (3.5)$$

$$L_A = E[\log(P(C_{\text{age}} = \text{age}|X_{\text{real}}))] + E[\log(P(C_{\text{age}} = \text{age}|X_{\text{fake}}))] \quad (3.6)$$

The training objective of Age-ACGAN's discriminator is to maximize $L_S + L_A$. This ensures the discriminator always maximizes the log likelihood of assigning the correct source of a CT image and its respective age class. On the other hand, Age-ACGAN's generator tries to maximize $L_A - L_S$. This is because despite the idea of fooling the discriminator with fake images (minimization of L_S), the classification accuracy of Age-ACGAN's auxiliary classifier has to be maintained (maximization of L_A).

3.2.3 Pixel Normalization

The concept of pixel normalization is used in PGANs (Progressive Growing of GANs) to normalize signal magnitudes. Unlike batch normalization, pixel normalization (PixelNorm) layers do not have trainable parameters and every pixel in pre-activation layers is normalized to unit length. PGAN only uses PixelNorm layers in its generator, whereas PixelNorm layers are used in both Age-ACGAN's generator and discriminator along with SELU layers to form PixelNorm-SELU blocks. Below are the mathematical definitions of pixel-wise normalization:

$$o_{x,y} = i_{x,y} \times \left(\frac{1}{C}\sum_{j=0}^{C} i_{x,y}^{j} + \epsilon\right)^{\frac{-1}{2}} \quad (3.7)$$

where i represents the input pre-activation with pixels at coordinates (x, y) and o represents the output layer. C is the number of channels (e.g. for a color image C=3: R, G, B and for grayscale image C=1). The small term ε is used to prevent zero-division

and is used in other normalization methods such as batch normalization (Equation 3.3) as well.

3.2.4 Minibatch Discrimination

An important feature of Age-ACGAN which contributed to its success is minibatch discrimination. Minibatch discrimination is an important heuristic used in other GAN architectures for maintaining stability and diversity in training. To understand how minibatch discrimination works, we first need to understand why mode collapse happens. As mentioned in previous sections, mode collapse happens when the generator 'collapses' onto a single or a few classes in the data distribution to 'fool' the discriminator. The outputs produced by a collapsed generator are usually high-quality but are extremely limited in variety. Minibatch discrimination tackles this problem by disallowing GANs to produce low-entropy solutions Below is the general workflow of minibatch discrimination:

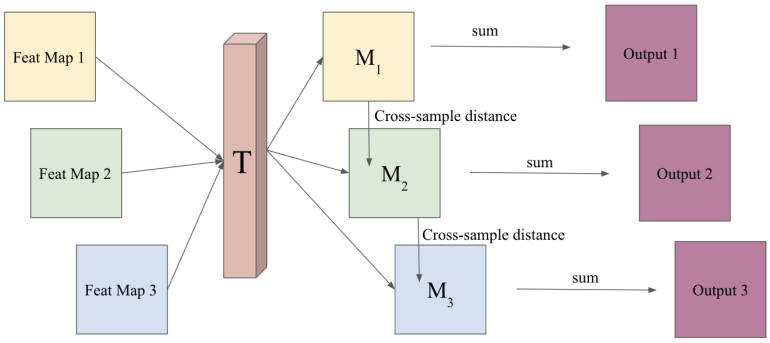


Figure 3.5 How minibatch discrimination works. Feature maps are multiplied by a fixed tensor and the cross-sample distances of the resulting matrices are computed.

Fig. 3.5 shows the implementation of minibatch discrimination. Minibatch discrimination is inspired by the success of batch normalization described in section 3.1.2. Each feature

map/ vector is multiplied by a fixed tensor T, and the resulting matrices are used to calculate cross-sample distance. Any distance metric can be used for cross-sample distance calculation, but the original authors choose to compute the L_1-distance across each row of a given matrix M. Finally, the output of the mini-batch discrimination layer is computed as follows:

$$o(x_i)_{batch} = \sum_{j=1}^{n} c_{batch}(x_i, x_j) \in \Re \tag{3.8}$$

where o represents the output, c represents the cross-sample distance metric and i, j represent row and column respectively. The outputs are finally concatenated to the original input feature maps and are passed on to the subsequent layers in the discriminator. Now the discriminator can distinguish and capture the side information of individual batches, and therefore able to identify and avoid low entropy solutions.

It may not seem obvious, but the addition of minibatch discrimination layers will not change the overall learning objective of any GAN. Recall the original training objective of GAN is to minimize the Kullback-Leibler (KL) divergence between two probability distributions P (training data) and Q (synthesized data), where:

$$D_{KL-div}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx \tag{3.9}$$

Observe that Equation 3.9 will yield $D_{KL\text{-}div} = 0$ when the two distributions are exactly the same ($\log(1) = 0$). We can now define new distributions P^n and Q^n to represent minibatch discrimination with a common sample size n:

$$P^n = P(x_1) \cdot P(x_2) \cdot \ldots \cdot P(x_n) = \prod_{i=1}^{n} P(x_i) \tag{3.10}$$

$$Q^n = Q(x_1) \cdot Q(x_2) \cdot \ldots \cdot Q(x_n) = \prod_{i=1}^{n} Q(x_i) \tag{3.11}$$

and now we substitute the distributions back into Equation 3.9:

$$D_{KL-div}(p||q) = \int p(x) \log \frac{\prod_{i=1}^{n} P(x_i)}{\prod_{i=1}^{n} Q(x_i)} dx$$

$$= \int p(x) \log \frac{P(x_1)}{Q(x_1)} dx + \ldots + \int p(x) \log \frac{P(x_n)}{Q(x_n)} dx$$

$$= n \cdot \int p(x) \log \frac{P(x_n)}{Q(x_n)} dx = n \cdot D_{KL-div}(p||q) \quad (3.12)$$

and here we show that by changing the learning objective to accommodate minibatch discrimination, the overall learning objective to minimize KL-divergence is not modified at all.

3.2.5 Residual Blocks

Residual blocks or residual connections in generate allow faster forward propagation through multiple layers of the network. Age-ACGAN uses residual blocks with a scalable parameter which determines the number of residual blocks. Below are two figures which show the differences between a regular block and a residual block:

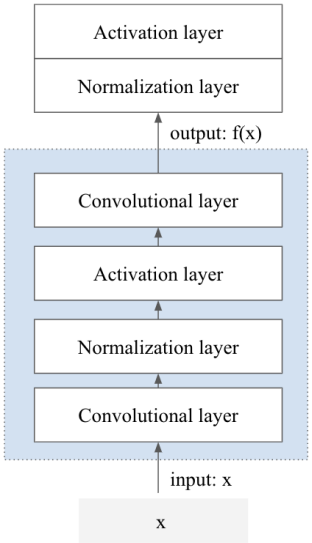


Figure 3.6 Schematic of a regular convolution block. The input x maps directly to the output f(x).

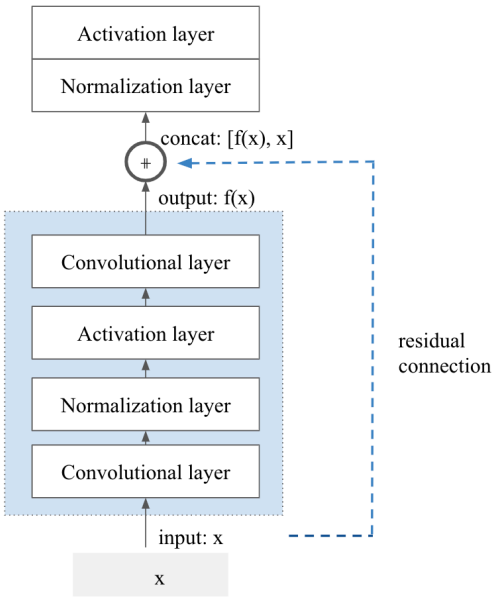


Figure 3.7 Schematic of a residual block. The input x is concatenated to the output f(x).

Typically, multiple layers are grouped together as 'blocks' in dCNNs. A residual block, as suggested by its name, concatenates its input to its output such that a residual mapping is learned by the block. As shown in Figure 3.6, a typical convolution block takes in input x and produces an output f(x). On the other hand, a residual block concatenates its input x to its residual mapping f(x) to produce an output [f(x), x].

3.2.6 Network Architecture

Even though the training objectives of Age-ACGAN and the original ACGAN are similar to each other, Age-ACGAN's network architecture has been heavily modified to adapt to medical image synthesis. Age-ACGAN's generator consists of 11 total convolutional blocks plus 7 residual blocks. We choose to use wider convolutional layers (up to 700 channels) and smaller kernel sizes (3x3) than the original ACGAN architecture. One residual block consists of two convolutional layers, two normalization layers (PixelNorm) and two activation layers (SELU).

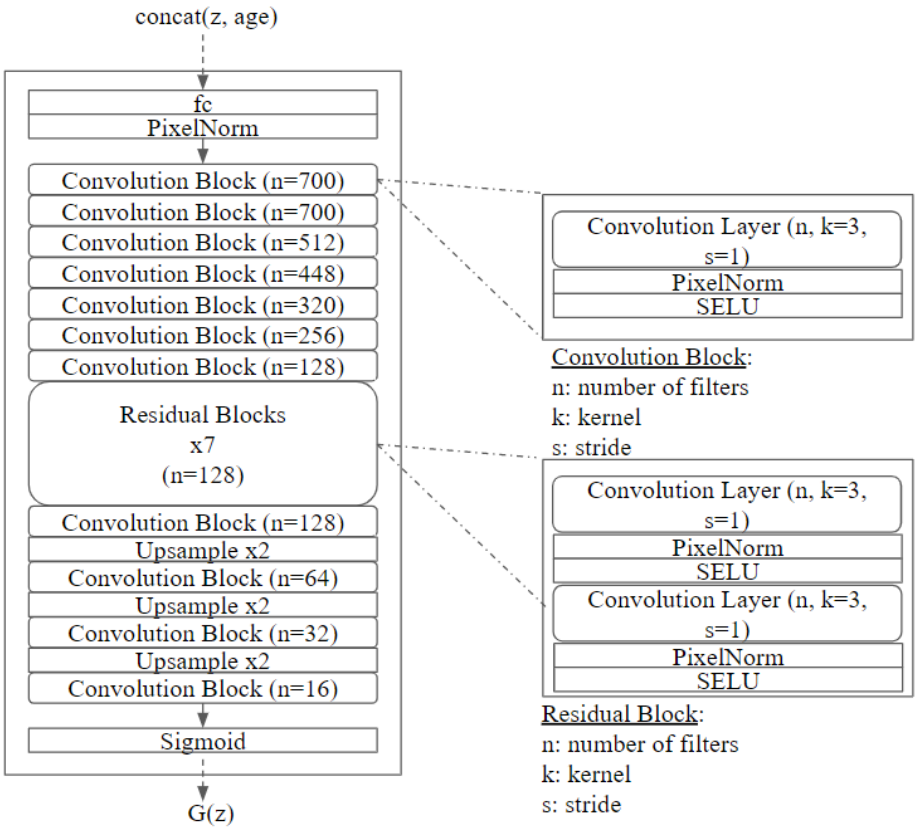


Figure 3.8 Age-ACGAN generator architecture. 7 residual blocks are used in addition to 11 convolutional blocks.

As mentioned in the previous sections, residual blocks with skip connections allow the construction of deeper networks without any gradient degradation issues. Moreover, the combination of PixelNorm and SELU layers increases network convergence speed in a fashion like the BS layers in our proposed BS-DCGAN.

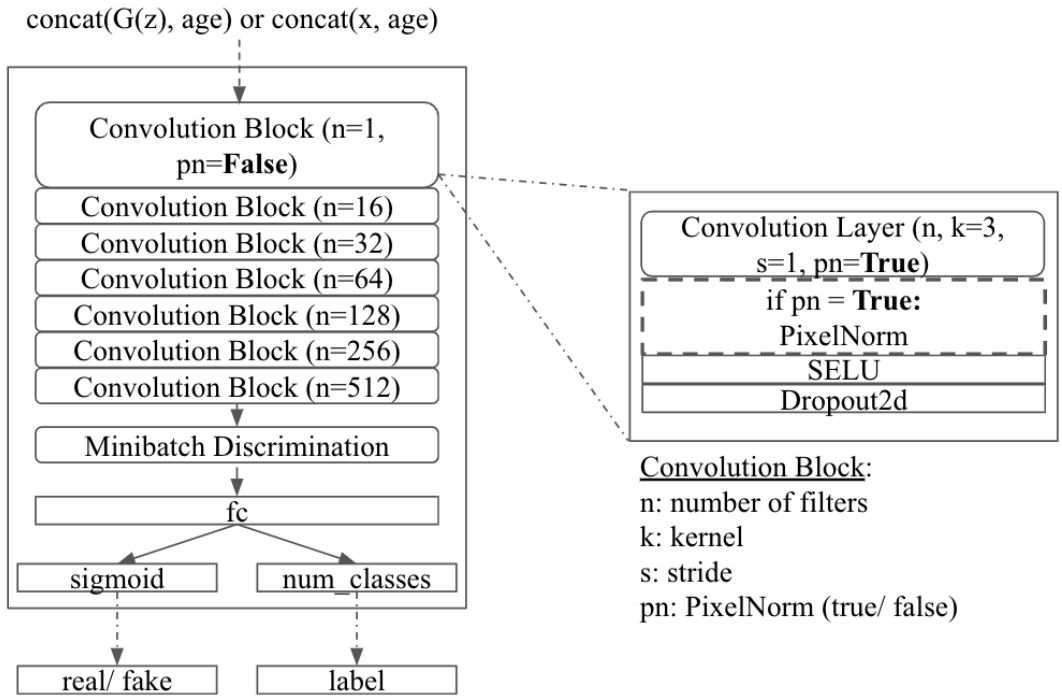


Figure 3.9 Age-ACGAN discriminator architecture. An additional auxiliary classifier is added to predict class labels along with the source of an input image

In contrast, Age-ACGAN's discriminator is like the original ACGAN's discriminator except the addition of a single convolutional layer (with 512 filters) and the aforementioned minibatch discrimination layer. All batch normalization layers and ReLU layers are switched to PixelNorm and SELU layers, respectively.

CHAPTER 4
CFG-SEGNET: A FEATURE-GENERATING FRAMEWORK FOR PEDIATRIC ABDOMINAL CT SEGMENTATION

4.1 Background and Training Objectives

4.1.1 Overview

The main goal of CFG-SegNet is to provide a unified framework to simultaneously generate novel training images and to learn an organ segmentation task. The proposed framework is made up of two networks, namely an Age-ACP2P (Age Auxiliary Classifier Pix2Pix) network and a U-Net. While the U-Net is responsible for segmentation, Age-ACP2P generates an extra batch of 'latent' features at each training iteration. It is important to mention that both Age-ACP2P and U-Net are trained on the same loss function, in which we will describe in detail in the following sections.

4.1.2 Loss Function

Age-ACP2P, the feature generation network of our proposed framework is considered to be one of the many conditional GAN (cGAN) variants. We choose to use the Pix2Pix network, a type of cGAN commonly used in image-to-image translation tasks as our baseline model. The loss function of Pix2Pix (Equation 2.4) is a combination of conditional adversarial and reconstruction losses. L_1 loss is generally preferred over L_2 loss for image reconstruction tasks since L_2 loss has quite a few poor implicit assumptions, such as assuming there is zero dependency between a given image and its noise. Note that we are free to replace the first term, the conditional adversarial loss term L_{cGAN} with the adversarial loss proposed in auxiliary classifier GANs (ACGANs) to incorporate class information from the training images. Specifically, we can substitute

L_{cGAN} with $L_{Age\text{-}ACGAN,}$ the aforementioned Age-ACGAN's conditional loss defined by Equation 3.5 and Equation 3.6:

$$G^* = \text{argmin}_G \text{max}_D L_{Age-ACGAN}(G, D) + \lambda L1 \quad (4.1)$$

Since our proposed CFG-SegNet jointly trains both a GAN and a U-Net, we have to incorporate the U-Net's segmentation loss in Equation 4.1 as well. There are many segmentation losses available including binary cross-entropy loss and dice loss. In our empirical studies, we found binary cross-entropy (BCE) loss produces better qualitative results than soft dice loss. BCE loss is summarized below as Equation 4.2:

$$L_{BCE} = \frac{-1}{N}\sum_{n=1}^{N}[y_n \log(h_\theta(x_n)) + (1 - y_n)\log(1) - h_\theta y_n] \quad (4.2)$$

where N is the size of the training data and h_θ is the segmentation model h with weights θ. Input and target labels for a particular training sample are represented by x_n and y_n respectively. Now we can include the segmentation loss defined above as an additional loss term in Equation 4.1, and yield CFG-SegNet's objective function:

$$G^* = \text{argmin}_G \text{max}_D L_{Age-ACGAN}(G, D) + \lambda_{L1} L1 + \lambda_{BCE} L_{BCE} \quad (4.3)$$

The two λs in the proposed objective function control the ratio between segmentation and image reconstruction losses. If $\lambda_{BCE} = 0$, our training objective will become the same as Age-ACP2P's training objective defined by Equation 4.1.

4.2 Implementation of CFG-SegNet

4.2.1 Channel-wise Concatenation of Age Class Labels

Age-ACP2P is the most important component of CFG-SegNet since it is responsible for generating novel training features in each iteration. As suggested by its name, Age-ACP2P is a combination of two conditional GANs: pix2pix and Age-ACGAN. Similar to pix2pix, Age-ACP2P uses a U-Net as its generator and a

convolutional “PatchGAN” classifier as its discriminator. However, instead of conditioning only on an input image x, Age-ACP2P’s generator also takes in age class labels as additional side information. Age class labels are appended as an extra channel to single-channeled (grayscale) input images to create 2-channel images. Below is a figure to help illustrate the concept of channel-wise concatenation of age class labels:

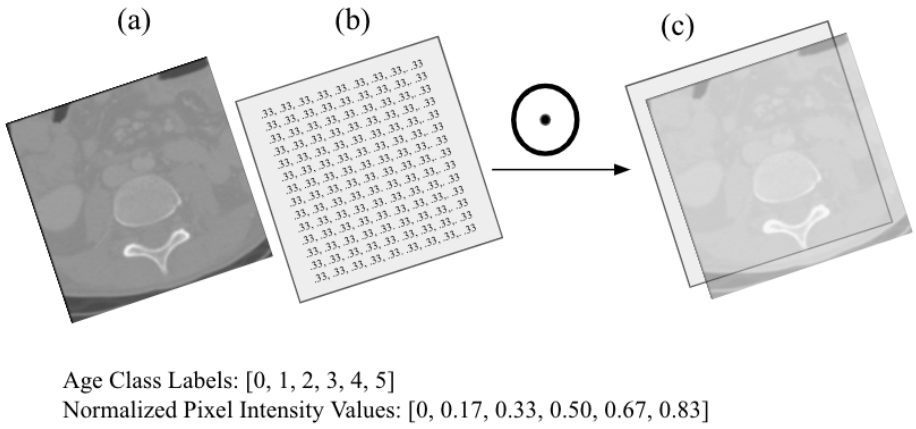


Figure 4.1 Channel-wise concatenation of age class labels. Age class labels are normalized as pixel intensity values and added as a second channel to a grayscale image to form 2-channel images.

As illustrated in the figure above, the age class labels are normalized as pixel intensity values arrays. For example, a patient who falls within age class 1 (we will further define age classes in the following sections) will have a 512 × 512 × 1 matrix filled with the same pixel intensity value $\frac{1}{6} = 0.17$ (**b**) concatenated to its original image (**a**) to form a 512 × 512 × 2 image (**c**).

4.2.2 Framework Design

The proposed framework consists of two networks, a U-Net segmentation network and a feature-generating Age-ACP2P. During the initialization of training, the U-Net first generates a segmentation mask f(x) given an input image x. The segmentation mask is then translated back into its latent feature x' by Age-ACP2P's generator. Since the translated feature x' is different from yet containing similar information as our original input image x, we can then retrain the U-Net to perform segmentation on x' to

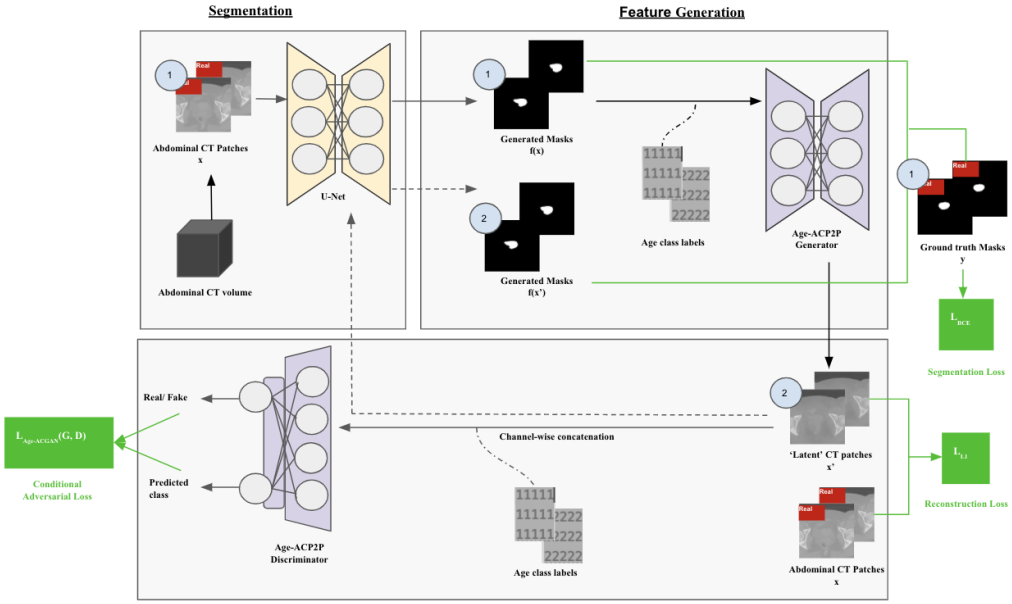


Figure 4.2 Workflow of CFG-SegNet. The 3 loss terms in our custom loss function are highlighted in green.

produce its respective segmentation mask f(x'). Moreover, since both networks are jointly trained by the same loss function, we expect the segmentation accuracy of U-Net to improve overtime as the quality of the translated features generated by ACP2P improves. Only the U-Net is needed for testing, and hence ACP2P can be viewed as an auxiliary trainer which aids U-Net during training. CFG-SegNet also works best when patches of CT images are used instead of the entire image. This is because unlike full

semantic maps, organ(s) segmentation masks only contain information about localized regions of the CT scan. Hence, center-cropping around the organ(s) being segmented is recommended for CFG-SegNet.

4.2.3 Atlas-based Localization in Testing

We use an atlas-based localization method in testing to identify organ locations in unseen CT images. Assuming the organ locations are known in the training data and are unknown in testing data, we can create image registrations using affine transformations (affine registration) to map out the organ locations in testing volumes. Specifically, for each age class we use a single CT volume in our training data as an age-conditioned atlas to extract organ locations in testing CT volumes of the same age class. Figure 4.3 below illustrates the general workflow of atlas-based organ segmentation and localization. Using an existing Insight Toolkit (ITK)-based image registration toolbox such as SimpleElastix [71], image registration can be created for a target CT volume given a fixed atlas. Transformation of the atlas's volumetric mask based on the image registration will produce an augmented mask of the target CT volume. Once the volumetric mask of the target CT volume is extracted, its center distance from the original atlas volumetric mask can be computed. Judging from the center distances and bounding coordinates of

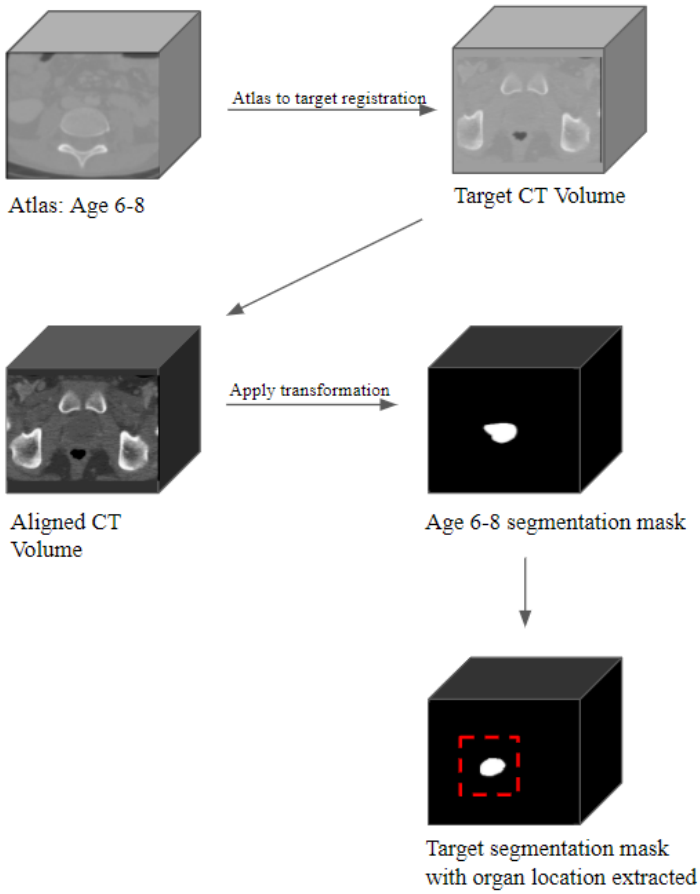


Figure 4.3 An Example of atlas-based organ segmentation and localization. Affine registration is used to transform an atlas mask to produce segmentation for the target volume. A smaller volume (highlighted in red) is cropped around the proposed organ location.

the target and reference masks, we can determine whether the extraction is successful. It is important to note that affine registration is chosen for this atlas-based preprocessing step since it has less degrees of freedom than deformable/ non-rigid registration.

CHAPTER 5
EXPERIMENTS AND RESULTS

5.1 Medical College of Wisconsin Pediatric Abdominal CT Dataset

5.1.1 Overview

All of the experiments described in the following sections use a pediatric abdominal CT dataset collected by the Medical College of Wisconsin in 2017. Lightspeed VCT CT system, designed and manufactured by General Electric (GE) Healthcare [72] is used for all imaging in this study. The dataset contains a total of 120 CT volumes representing 64 male and 56 female patients ranging from ages 1 to 17. Figure 5.1 summarizes the age distribution for both sexes, where both males and females have a median age of 8.

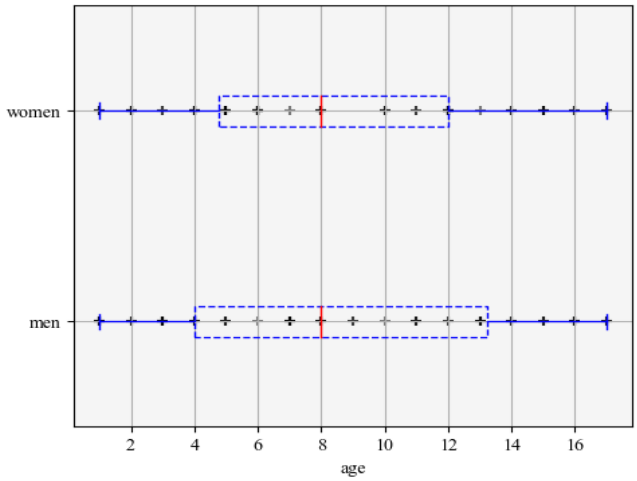


Figure 5.1 Age distribution of MCW pediatric abdominal CT dataset. Both men and women have an equal distribution in age

For each patient, there are 25 possible volumetric segmentation masks hand-drawn by radiologists at MCW. Several organs such as the uterus, the prostate and the breasts are

specific to either the males or the females. Each patient has an average of 16 to 17 available organ segmentation masks.

- Adrenal glands (left and right)
- Bladder
- Breast (left and right)
- Duodenum
- Esophagus
- Femoral head (left and right)
- Gallbladder
- Gonads
- Heart
- Kidneys (left and right)
- Large intestine
- Liver
- Pancreas
- Prostate
- Rectum
- Small intestine
- Spinal canal
- Spleen
- Stomach
- Thymus
- Uterus

Availability of hand-drawn organ segmentation masks for each organ is detailed in Figure 5.2. Availability for reproductive/ gender-specific organs such as breasts, prostate and uterus are relatively low compared to common organs such as liver and heart.

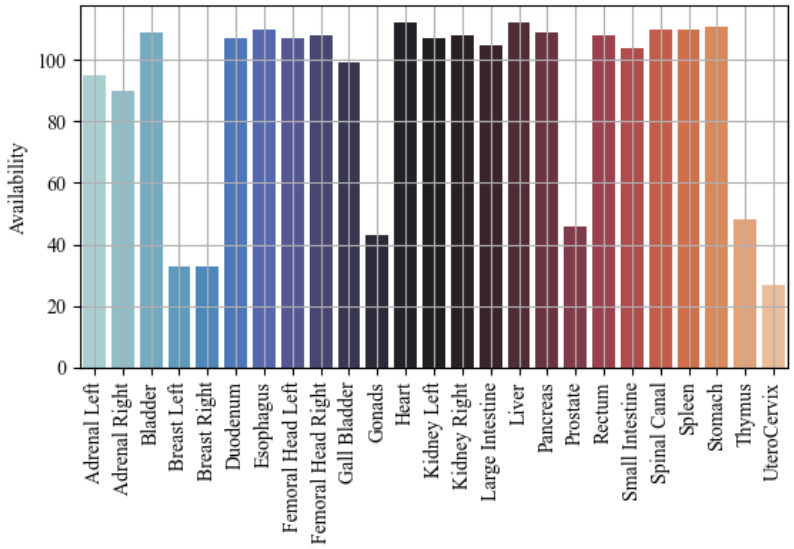


Figure 5.2 Organ availability of MCW pediatric abdominal CT dataset. The following organs have the lowest availabilities: breasts, gonads, prostate, thymus, uterus

Most patient identifiers are removed via XNAT Edit Script [73]. However, certain header information such as the patient's sex and birthdate are kept for research purposes. Other important unique identifiers and sensitive information such as patient ID, physician's name, operator's name and institution name are either jittered or removed from the file header.

5.1.2 File Format

All images are stored as DICOM (Digital Imaging and Communications in Medicine) files and patient information is stored in DICOM headers (see section 2.4.2 for

more details about DICOM format and file headers). Each DICOM file represents a single-channel, 2-dimensional slice of the CT volume, and all images have the same dimensions (512 × 512 × 1). A given patient has a number of 2D slices ranging from anywhere between 500 to more than 1000 slices. These slices can be concatenated along the z-axis to form a single, 3D image volume. Image volumes can be parsed and shown in visualization interfaces such as MIPAV [74] or Slicer3D [75] as shown below:

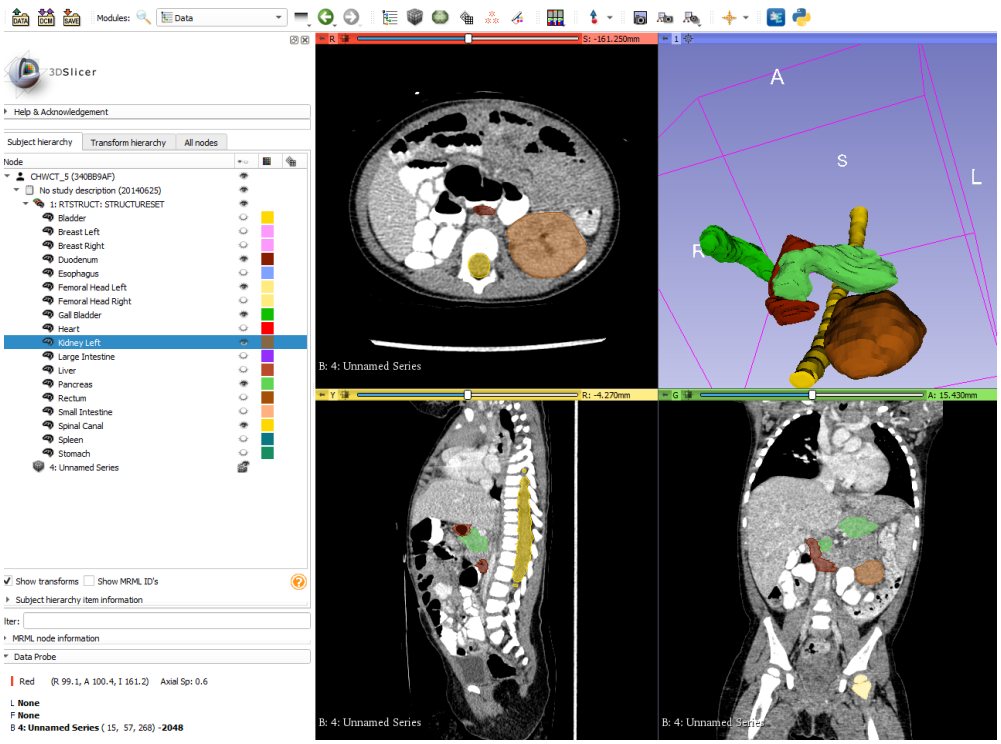


Figure 5.3 Visualization interfaces for medical images. Above is a screenshot from Slicer3D, which is commonly used to visualize DICOM/ NIfTI image volumes

MCW pediatric abdominal CT dataset's organ segmentation masks are stored in DICOM-RTSTRUCT format [76]. RTSTRUCT stores a volume of RT dataset containing multiple contours and can be extracted with the aforementioned visualization tools. However, in order to conveniently extract desired organ segmentation masks a Python preprocessing script is written.

5.2 Unconditional Image Synthesis with BS-DCGAN

5.2.1 Experimental Design

We use 20 pediatric abdominal CT volumes from the MCW dataset to train both a DCGAN and our proposed BS-DCGAN. The 20 randomly selected patients range from ages 2 to 4, and a total of 424 2D slices containing the liver region are extracted. Full 512 × 512 CT images are used in this experiment to illustrate the ability of BS-DCGAN to synthesize viable, high-resolution CT images.

Both GAN architectures are trained for at least 200 epochs to ensure sufficient convergence in the generator and the discriminator. Each model's generator and discriminator weights on the 200th epoch is saved and used to subsequently generate 16 synthetic CT images from random noise. It is important to note that the same random noise vector is used in both architectures to eliminate possible confounding factors.

Multi-scale structural similarity index measure (MS-SSIM) score [77] is used as a metric to evaluate the quality of the images generated from both architectures. Structural similarity index measure (SSIM) is a quantitative measure which evaluates the similarity between two images. Both MS-SSIM and SSIM range from 0 to 1, and two given images are exactly the same when their MS-SSIM/ SSIM is 1. SSIM is defined as follows:

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_x^2 + c_2)} \tag{5.1}$$

where μis the average, $\sigma_{xy}$is the covariance and c_1 and c_2 are constants. As suggested by its name, MS-SSIM evaluates the SSIM scores of two images at various scales via downsampling and applying a low-pass filter. MS-SSIM is defined as follows:

$$\mathrm{SSIM}_{MS}(x, y) = [l_M(x, y)]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \tag{5.2}$$

where $l_M(x,y)$ is the luminance comparison between two images x and y at the highest possible scale M. α, $\beta$and $\gamma$are parameters which can be adjusted to tune the importance of various components in Equation 5.2. Since MS-SSIM is more generalizable compared to single-scale methods such as the original SSIM, we choose MS-SSIM as our evaluation metric.

5.2.2 Results

Both models converge after 200 epochs. However, upon close examination of validation results during training mode collapse is found in DCGAN. Figure 5.4 shows an example of full-resolution synthetic CT generated by DCGAN and our proposed BS-DCGAN. BS-DCGAN-generates images that are apparently higher in resolution, containing more complex features and are less noisy than DCGAN-generated images. Since DCGAN experiences mode collapse after training for 200 epochs, its generated images lack diversity and are almost identical.

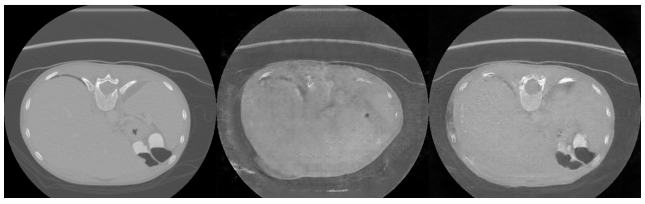


Figure 5.4 Synthetic images generated by DCGAN (middle) **and BS-DCGAN** (right) **compared to a real CT image** (left). The abdominal CT slice generated by BS-DCGAN seems more realistic than the one generated by DCGAN.

Besides being capable of generating high quality liver CT images of pediatric patients, BS-DCGAN is also capable of generating images that are structurally similar to the ground truth clinical images in the MCW abdominal CT dataset. Pairwise MS-SSIM scores are calculated between images generated by the two trained models and real images from the dataset. On average, BS-DCGAN has a MS-SSIM score of 0.720 across

Table 5.1 Average MS-SSIM of synthetic CT images. BS-DCGAN has a higher MS-SSIM when compared to ground truth images

$\textbf{MS-SSIM}_{\textbf{DCGAN}}$	$\textbf{MS-SSIM}_{\textbf{BS-DCGAN}}$
0.677	0.720

16 images, while DCGAN has 0.677. Statistical significance testing (two-sample t-test) confirms that BS-DCGAN indeed has a higher average MS-SSIM score in its generated images (p-value: 0.003, α: 0.05).

5.3 Conditional Image Synthesis with Age-ACGAN

5.3.1 Experimental Design

We design an experiment to generate 172 × 172 2D patches center-cropped around the pancreas along their segmentation labels with Age-ACGAN. 5 patients from each of the following 3 age-classes (15 patients total) are selected from the MCW pediatric abdominal CT dataset:

- Infant class (age 1 to age 3)
- Preschool class (age 4 to age 6)
- Adolescent class (age 16 to 18)

20 abdominal CT slices containing the pancreas are selected as training data for each of the 15 patients, and 172 × 172 patches are created via center-cropping around the pancreas in both the CT and its respective organ label. Original resolution is preserved in our training data since there is no downsampling, and patches are simply cropped from the original images. No additional localization is needed since the organ locations are known in our segmentation label map. To save time, CT patches and label patches are

zero-mean whitened and concatenated together as 344 × 172 input images to our proposed Age-ACGAN during preprocessing. In both training and testing phases, age class labels are concatenated to random Gaussian noise vectors z before inputting to both the generator and the discriminator of Age-ACGAN. Cross entropy is used in our implementation to calculate Age-ACGAN's loss terms L_s and L_A defined in Equations 3.5 and 3.6. An arbitrary cutoff point α=0.5 is also used to binary threshold synthesized masks to pixel intensity values 0 and 1. Additionally, we train a DCGAN on the same set of images as baseline comparison. Image synthesis with DCGANs is unconditional and hence no additional age class labels are required. We train both networks with a fixed batch size of 16 for 3,000 epochs.

While examining the training images, we also observe the pancreas to vary anatomically as a patient's age increases. Specifically, the pancreas becomes more elongated in shape as the patient grows older. Figure 5.5 are three images sampled from

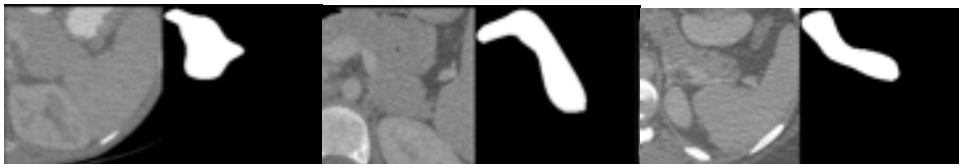


Figure 5.5 Sample training images From Infant class (left), **Preschool class** (middle) **and Adolescent class** (right). Notice the pancreas becomes more elongated as patient age increases.

training dataset to illustrate this growth trend. As shown in Figure 5.5, the pancreas becomes more elongated as patients grow older. However, it is important to note that this trend is only observable in the MCW pediatric abdominal CT dataset and should not be generalized without further investigation.

5.3.2 Results

We study the convergence of generator loss during training. While Age-ACGAN quickly converges after 500 iterations during training, DCGAN fails to converge even after 1000 iterations of training and experiences constant increase in loss (Figure 5.6). Age-ACGAN also generates CT patches along with their pancreas masks in higher quality and contains finer details than those generated by the DCGAN. While both networks do not experience mode collapse, many of the images generated by the DCGAN contain shape and streak artifacts (Figure 5.7). Images generated by the

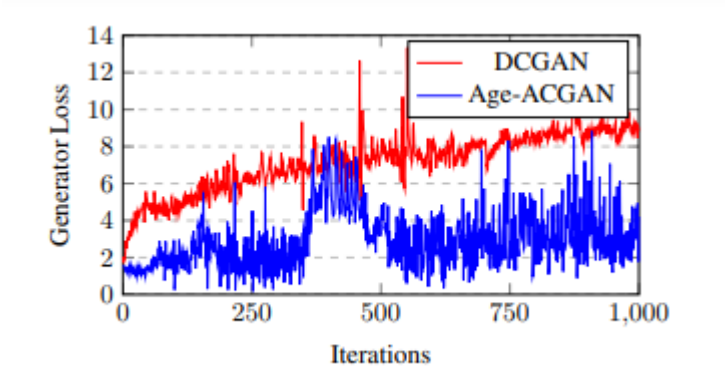
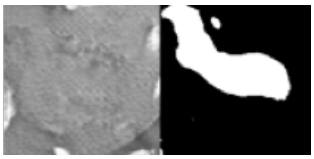
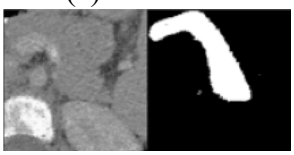


Figure 5.6 Generator loss of Age-ACGAN (blue) **and DCGAN** (red). Age-ACGAN quickly converges around the 500th iteration, while DCGAN's generator loss continues to increase.

Age-ACGAN do not have any streak artifacts yet contain a low level of natural CT noise and artifacts, such as ring artifacts and cone beam artifacts. More importantly, DCGAN is unable to capture the precise shape of the pancreas in patients of varying ages. Age-ACGAN, on the other hand, is able to capture the aforementioned growth trend in its synthesized images due to age class label conditioning and minibatch discrimination (Figure 5.8).



(a) DCGAN



(b) Age-ACGAN

Figure 5.7 Pancreas CT and organ label generated by (a) **DCGAN and** (b) **Age-ACGAN**. Age-ACGAN is capable of generating high-quality CTs and pancreas labels.

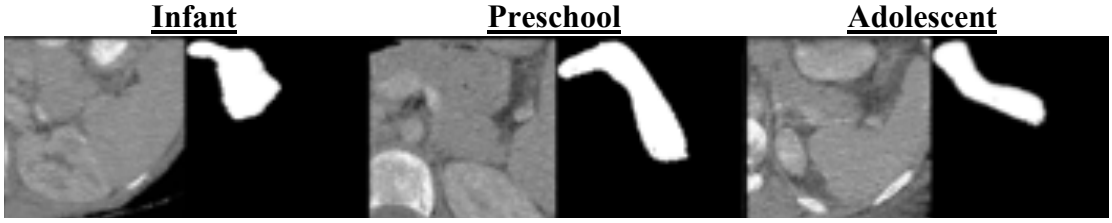


Figure 5.8 Sample pancreas CT and organ label generated by Age-ACGAN from each age class. Notice the pancreas label follows the same growth trend in the training images.

5.4 Organ Segmentation with CFG-SegNet

5.4.1 Experimental Design

There are very few publicly available pediatric abdominal CT datasets containing reproductive organs such as the prostate and the uterus. To investigate the ability of our proposed CFG-SegNet to segment these hard-to-find organs, we design 2.5D segmentation experiments which use a total of 64 abdominal CT volumes from the MCW pediatric abdominal CT dataset. Out of the 64 patients' image volumes, 24 volumes represent females containing the uterus and 40 volumes represent males containing the prostate. Each patient's CT volume is preprocessed into batches of 2D slices normalized by the mean and the variance of the volume. The produced segmentation mask for each 2D CT slice of a given patient is concatenated along the z-axis to form a single,

volumetric mask for evaluation. We carry out a total of 2 experiments: in the first experiment, CFG-SegNet, CE-Net (Context-Encoder Network) and U-Net are trained to perform 2.5D segmentation on 24 female CT volumes containing the uterus. A 4-fold cross-validation is used, with each fold containing a 75%-25% train-test split. Both training and testing volumes are preprocessed into 2D slices in the first experiment and are subsequently center cropped into 256 × 256 patches where the original image resolution is preserved.

In our second experiment, we perform segmentation of the prostate in a similar fashion as the first experiment for baseline analysis. We then incorporate an atlas-based localization technique using affine registration to extract the location of the prostate in testing images and re-test the trained networks. As mentioned in section 4.2.3, testing is performed under the assumption that the location of an organ in the CT image is unknown in clinical settings. SimpleElastix is used to extract organ locations in testing volumes via affine registration. Since there exists a huge variation in the prostate's length along the z-direction, we sparingly set a 256 × 256 × 100 (x × y × z) bounding box around the center of mass of the proposed location of the prostate. Each bounding box is then subsampled into 100 256 × 256 2D slices as network input. Since the testing volumes contain varying pixel spacings, they are resampled to have uniform spacing and direction before affine registration is performed. Similar to the first experiment, a 4-fold cross-validation is used with each fold containing a 75%-25% train-test split. We train our proposed CFG-SegNet, along with a U-Net and an Attention U-Net [78] to perform segmentation on the 40 patient CT volumes containing the prostate.

In both experiments, segmentation performance is evaluated with 2 well-established metrics: Dice Similarity Coefficient (DSC) and Hausdorff Distance (HD). Dice Similarity Coefficient, also known as the Sorensen-Dice Similarity Coefficient, is a statistic commonly used to measure the similarity of 2 images (or volumes) by computing their spatial overlap:

$$\mathrm{DSC} = \frac{2|A \cap B|}{|A| + |B|} \tag{5.3}$$

where A and B are 2D or 3D images, and A ∩ Brepresents their intersection. Similar to the similarity metric Intersection over Union (IoU), DSC ranges from 0 to 1 with 1 being a perfect match between the two images. On the other hand, Hausdorff Distance is the maximum distance of a set (or in our case, an organ segmentation label) to the nearest point in another set. In other words, two images with a shorter HD are more similar than two images with a higher HD. HD can be implemented with any distance metric, such as the Euclidean Distance. Since the calculation of HD requires two established sets, we only calculate the HD when a segmentation mask is available for a given CT image.

An important attribute of CFG-SegNet is the incorporation of age class labels. Similar to our previous experiment with the Age-ACGAN, we classify a patient's age into one of the following classes:

- Infant class (age 1 to age 3)
- Preschool class (age 4 to age 6)
- School age I (age 7 to age 9)
- School age II (age 10 to age 12)
- Adolescent I (age 13 to age 15)
- Adolescent II (age 16 to age 17)

In both experiments, each age class is approximately represented by the same number of patients. However, the number of patients in the Infant class is relatively lower than the other classes because of a lack of available training volumes from the original dataset.

5.4.2 Results

The first experiment benchmarks the proposed CFG-SegNet against two other segmentation networks, in particular the U-Net and the CE-Net. CE-Net has achieved state-of-the-art segmentation results on multiple medical imaging datasets including lung CT segmentation. Its success can be attributed to the addition of dense atrous convolution (DAC) and residual multi-kernel pooling (RMP) blocks, which aim to better preserve spatial information than the traditional U-Net. All 3 networks are trained for a total of 50 epochs to ensure convergence, and the saved model weights of the final epoch are used for testing. The computed DSC and HD averaged across 4-fold cross-validation is reported below:

Table 5.2 Mean uterus segmentation results with our proposed CFG-SegNet, CE-Net and U-Net. Best results are highlighted in bold, and all values shown are the average result of a 4-fold cross-validation experiment.

	DSC	**HD**
CFG-SegNet	**0.724 ± 0.0413**	**0.709 ± 1.56**
CE-Net	0.706 ± 0.0415	1.08 ± 1.67
U-Net	0.697 ± 0.0419	1.09 ± 1.92

We qualitatively evaluate the results of our first 2.5D uterus segmentation experiment. As shown in Figure 5.9, CFG-SegNet is able to generate a more accurate uterus segmentation mask (though not perfect) for a 5-year-old patient than the other 2

networks. All 3 segmentation networks struggle to generate perfect uterus segmentation masks for patients in the Infant and Preschool classes. However, our proposed CFG-SegNet is able to approximate the correct size of the mask and has fewer false positives than the other 2 networks. Both CFG-SegNet and CE-Net are able to accurately predict

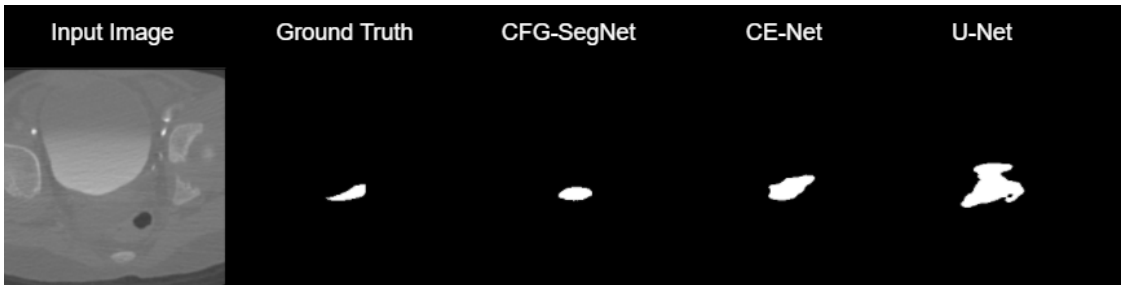


Figure 5.9 Sample uterus segmentation labels generated by CFG-SegNet, CE-Net and U-Net. Although none of the networks is able to produce a perfect segmentation for the uterus, CFG-SegNet appears to generate the closest approximation when compared to the ground truth.

uterus segmentation labels slice-by-slice for patients in the Adolescent I and Adolescent II classes since there is an abundance of adolescent training volumes available. We attribute the success of CFG-SegNet on correctly predicting infant segmentation labels to age-class labelling and ACP2P's feature-generating functionality.

For the second experiment, CFG-SegNet is benchmarked against the Attention U-Net- an advanced variant of the U-Net which uses attention gates (AG) on a subset of image volumes containing the prostate. Attention U-Net is previously used to segment the pancreas in 150 abdominal CTs, where it outperforms several state-of-the-art pancreas segmentation algorithms such as the Multi-Model 2D FCN and Hierarchical 3D FCN. Similar to the first experiment, all 3 networks are trained for a total of 50 epochs where the model weights for the final epoch are saved for testing. Testing is performed slice-by-slice on 256 × 256 × 100 image volumes extracted from center cropping (Table

5.3) and atlas-based localization (Table 5.4). Tables 5.3 and 5.4 summarize the computed DSC and HD values of our 4-fold cross-validation experiment with 40 CT volumes containing the pancreas with center cropping and atlas-based localization preprocessing. In both preprocessing methods, CFG-SegNet has the highest mean DSC and the lowest HD out of the 3 networks. We also examine the segmentation results for individual age classes, in particular the younger patients (i.e. patients that belong to the Infant, Preschool, School Age I and School Age II classes).

Table 5.3 Mean prostate segmentation results with our proposed CFG-SegNet, Attention U-Net and U-Net (center cropping). Best results are highlighted in bold, and all values shown are the average result of a 4-fold cross-validation experiment.

	DSC	**HD**
CFG-SegNet	**0.929 ± 0.200**	**0.338 ± 0.965**
Attention U-Net	0.925 ± 0.195	0.414 ± 1.21
U-Net	0.923 ± 0.167	0.390 ± 1.01

Table 5.4 Mean prostate segmentation results with our proposed CFG-SegNet, Attention U-Net and U-Net (atlas-based localization). Best results are highlighted in bold, and all values shown are the average result of a 4-fold cross-validation experiment.

	DSC	**HD**
CFG-SegNet	**0.769 ± 0.353**	**2.462 ± 0.965**
Attention U-Net	0.754 ± 0.367	2.64 ± 1.15
U-Net	0.7435 ± 0.350	2.75 ± 1.16

Figures 5.10 and 5.11 are paired class-wise boxplots which summarize the segmentation results (DSC) of our proposed CFG-SegNet and Attention U-Net. DSC values of background slices (negative samples that do not contain the prostate label) are excluded

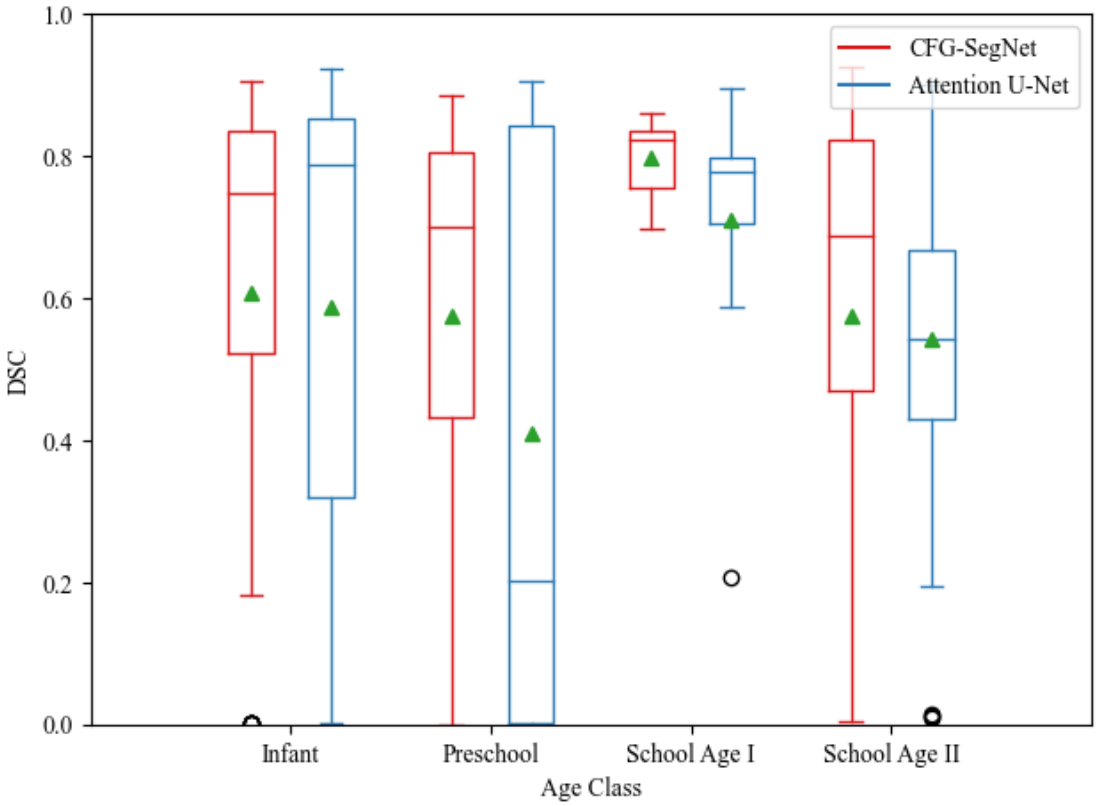


Figure 5.10 Paired class-wise boxplot of CFG-SegNet and Attention U-Net prostate segmentation results (center cropping). CFG-SegNet is better than Attention U-Net in all age classes.

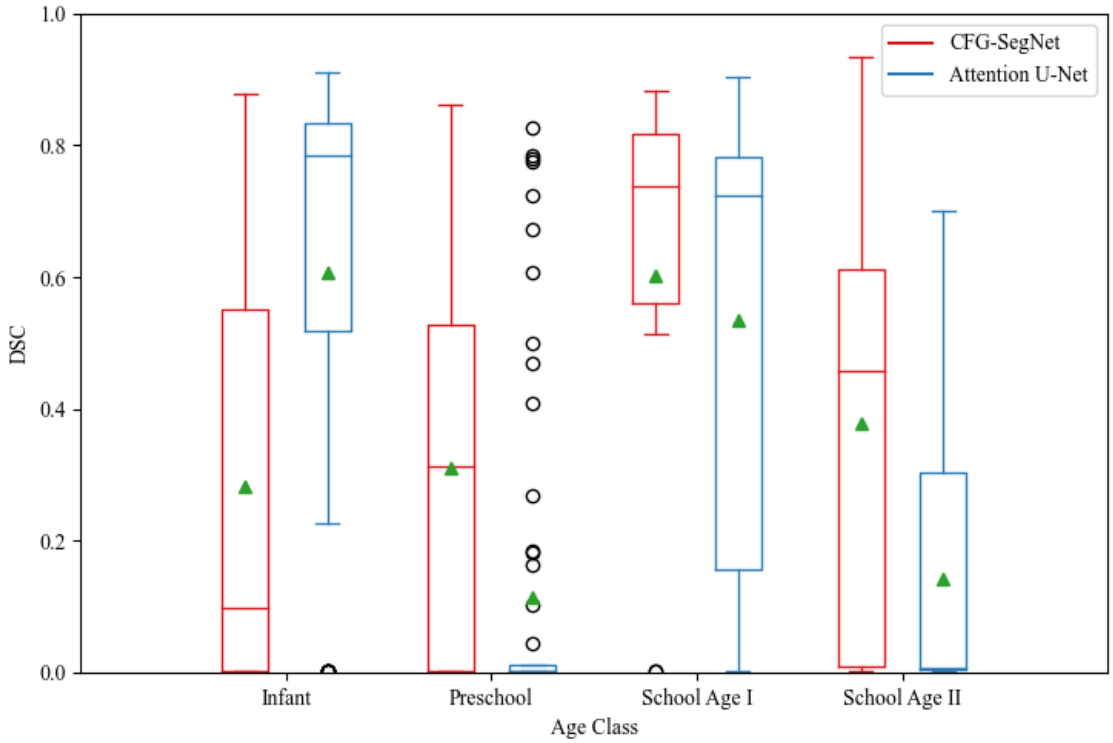


Figure 5.11 Paired class-wise boxplot of CFG-SegNet and Attention U-Net prostate segmentation results (atlas-based localization). CFG-SegNet is more consistent in segmentation performance than Attention U-Net across multiple age classes.

to make the results more meaningful. We suspect the large margin of error in our atlas-based preprocessing step leads to lower segmentation performance in young pediatric patients due to high variability in organ locations. Regardless of what preprocessing method is used, CFG-SegNet is more consistent in its segmentation results across the 4 age classes, whereas Attention U-Net struggles to produce segmentation masks for patients that belong to the Preschool class.

Finally, we evaluate the quality of the generated prostate masks via manual inspection. As shown in Figure 5.12, CFG-SegNet generates the closest approximation of the prostate mask when compared to the ground truth given the testing images are center-

cropped. A comparison of 3 prostate segmentations of a 5-year-old patient generated by CFG-SegNet, Attention U-Net and U-Net with atlas-based localization is provided in

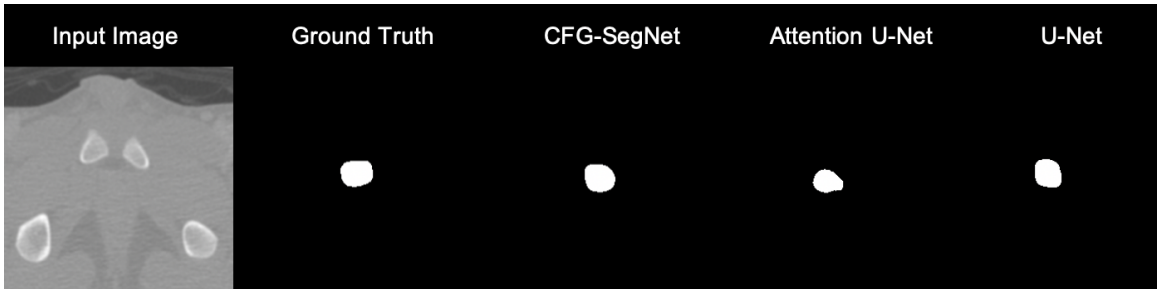


Figure 5.12 Sample prostate segmentations from CFG-SegNet, Attention U-Net and U-Net (center cropping). CFG-SegNet appears to generate the closest approximation when compared to the ground truth.

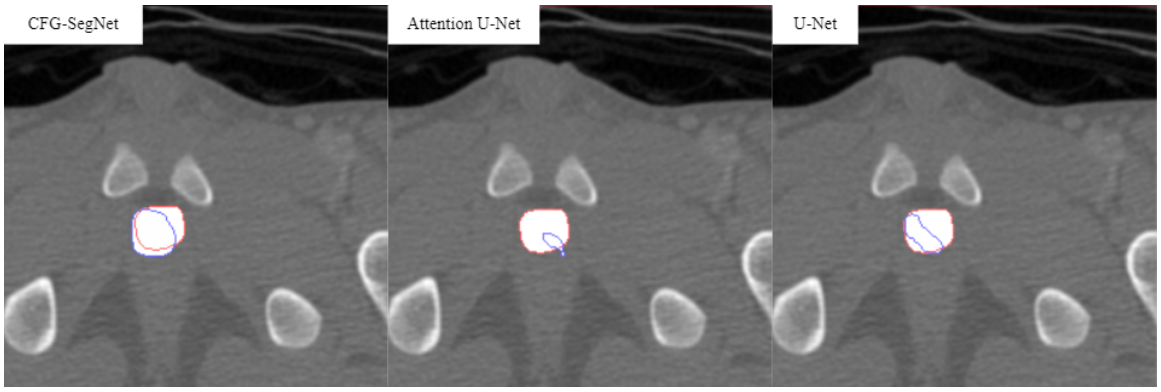


Figure 5.13 Sample prostate segmentations from CFG-SegNet, Attention U-Net and U-Net (atlas-based localization). Ground truth contours are highlighted in red, while the contours of the proposed labels are highlighted in blue.

Figure 5.13. Unlike Attention U-Net and U-Net, it is apparent that CFG-SegNet produces prostate segmentation masks for the 5-year-old patient with higher accuracy based on the highlighted contours. Besides being able to somewhat extract the prostate's location, Attention U-Net produces a relatively small segmentation mask that is considerably worse than the U-Net's segmentation. This is supported by the fact that Attention U-Net fails to produce meaningful prostate segmentations for patients between the ages of 4 to 6.

CHAPTER 6
DISCUSSION AND CONCLUSION

6.1 Summary of Major Contributions

In this thesis, we review the use of deep learning in medical image analysis. We examine the limitations with currently available methods in abdominal CT organ segmentation and propose 3 methods to overcome these limitations. A major roadblock in many organ segmentation algorithms is the lack of pediatric training data. Low availability of pediatric abdominal CTs leads to poor segmentation performance on organs that are either anatomically challenging or considered extremely radiosensitive (such as reproductive organs). Generative adversarial network (GAN) is a deep generative framework which makes use of two competing (adversarial) deep convolutional neural networks (dCNN) to generate realistic images from random noise. We propose the use of BS layers, which effectively combines batch normalization (BatchNorm) with scaled exponential linear units (SELU) in a deep convolutional generative adversarial network (DCGAN). Our proposed BS-DCGAN can generate abdominal CT images containing the liver at full resolution (512 × 512) and with high fidelity. Quantitative evaluation of the generated images with pairwise multi-scale structural similarity index measure (MS-SSIM) score show a high similarity between abdominal CTs generated by BS-DCGAN and ground truth CTs.

Next, we examine the possibility of synthesizing abdominal CTs along with their pancreas labels via the use of an auxiliary classifier generative adversarial network (ACGAN). The proposed Age-ACGAN network adapts the previously proposed BS layers with pixel normalization to increase convergence speed and improve training stability. Age-ACGAN also uses residual layers to aid gradient propagation throughout

the generator network and uses minibatch discrimination to ensure diversity in generated images. Comparison of convergence speed shows a faster convergence in Age-ACGAN's generator than DCGAN's generator, in which Age-ACGAN converges after the first 500 iterations of training. Age-ACGAN produces abdominal CT patches and pancreas labels of size 172 × 172, center-cropped around the pancreas from full-resolution (512 × 512) abdominal CTs. Unlike the images generated by the DCGAN, images generated by Age-ACGAN are free of streak artifacts and effectively capture the growth trend of pancreas from younger to older patients.

Finally, a unified segmentation framework which jointly trains a image-to-image translation GAN (Age-ACP2P) and a segmentation network (U-Net) is proposed. A new loss function is also proposed to combine adversarial loss with image reconstruction and segmentation losses. The proposed CFG-SegNet is tested with a subset of patient CT volumes of the MCW pediatric abdominal CT dataset which contains 2 reproductive organs: the prostate and the uterus. Both quantitative and qualitative results from our 2.5D segmentation experiments on CT volumes of 24 female patients and 40 male patients indicate that CFG-SegNet produces uterus and prostate segmentations with higher accuracy, and its segmentation performance is more consistent across patients in multiple age classes.

6.2 Progress of Current Work

All the work presented in this thesis is either published, under review or pending submission. BS-DCGAN is published as a one-page paper and poster presentation in 2019 IEEE 41st Annual International Engineering in Medicine and Biology Conference (EMBC). Age-ACGAN is published as a 4-page paper in the conference proceedings and

is selected for oral presentation at the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). CFG-SegNet is accepted to the 2020 Society of Photographic Instrumentation Engineers (SPIE) Medical Imaging Conference as a conference paper and is selected for oral presentation. A more detailed write-up of CFG-SegNet will be submitted to the IEEE Transactions of Medical Imaging (TMI) journal by the end of this year. A paper detailing the synthesis of multi-channel electroencephalogram (EEG) in the form of Gramian Angular Field (GAF) images with a Gramian Temporal Generative Adversarial Network (GT-GAN) is submitted to the International Conference on Acoustics, Speech, and Signal Processing 2021 (ICASSP 2021). Source code of BS-DCGAN is available at the author's repository (https://github.com/chinokenochkan/bs-dcgan). The remaining source codes will be open-sourced upon the publication of this thesis.

6.3 Limitations and Future Work

One of the fundamental assumptions of Age-ACGAN is that it assumes the availability of age information when it is often anonymized in real, clinical settings due to privacy issues related to electronic health records (EHRs). Moreover, there are other confounding factors such as race and underlying health conditions that may or may not affect the outcome of image synthesis. All 3 proposed networks in this thesis need to be further tested on a larger, publicly available pediatric dataset which contains multiple organ labels and accurate patient information. Such a dataset is currently not available, and thus the scope of our work is limited in this regard.

Like other supervised segmentation methods, our proposed CFG-SegNet is limited by the quality of the manually annotated labels. The proposed atlas-based

localization method is also affected by the inconsistencies in reference frames, which renders the normalization of patient coordinate systems from 2 different patients impossible during the image registration. The frame of reference is chosen arbitrarily during each scan, and it can be as simple as a fixed point on a scanner relative to the table where the patient lies. If this is kept constant throughout multiple patients, then we can align CT volumes from 2 patients with their respective image origins (the coordinates of the center of the first voxel).

In general, hardware limitation is trivial as we are provided with powerful deep learning workstations containing multiple RTX 2080 ti GPUs. Our proposed CFG-SegNet also contains fewer parameters than Attention U-Net. This is in fact quite impressive considering CFG-SegNet uses a conditional version of Age-ACGAN, which contains a modest number of parameters in itself. However, scaling up our proposed CFG-SegNet to volumetric segmentation will require 3D convolutions which in turn requires higher computing power.

One possible extension of this work is to combine CFG-SegNet with self-supervised learning. Self-supervised learning has gained popularity in recent years due to the issue with manually annotated labels. Annotated data is in fact quite costly, even more so in the field of medical imaging. Successful self-supervised learning frameworks commonly use representation learning to learn mappings in unlabeled datasets [80]. Representation learning can be achieved via patch location prediction, reconstruction of distorted image, or any other methods that withhold a portion of information from unlabeled images. If CFG-SegNet can be further improved to self-supervise, then the use of age information is indeed trivial. In the meantime, patient attributes other than the

patient's age can also be used to improve segmentation quality of CFG-SegNet. It will be interesting to modify our proposed Age-ACGAN or Age-ACP2P to generate CT slices conditioned on a patient's gender or even genetic information. Moreover, Age-ACP2P can also be adapted to translate various imaging modalities such as the conversion of CT and MR images. An attempt has been made to convert Age-ACP2P into a volumetric synthesis network similar to Vox2Vox [81], but the synthesized volumes are barely recognizable due to a huge discrepancy in performance between the generator and the discriminator (the generator is given a much harder task than the discriminator).

6.4 Conclusion

Accurately segmenting abdominal organs in CTs has always been a challenging task in medical image analysis. While many deep learning solutions exist for abdominal CT organ segmentation, their common requirement of large training datasets remains a challenge. More importantly, pediatric abdominal CT datasets are not readily and publicly available for training. In this thesis, we propose a total of 3 frameworks: BS-DCGAN (BatchNorm-SELU deep convolutional generative adversarial network) for unconditional abdominal CT synthesis, Age-ACGAN (Age auxiliary classifier generative adversarial network) for age-conditioned abdominal CT patch synthesis and CFG-SegNet (conditional feature generation segmentation network) for 2.5D patch-based reproductive organ segmentation in abdominal CTs. All 3 of our proposed approaches are thoroughly validated with image volumes from the MCW pediatric abdominal CT dataset. Experimental results comparing our proposed methods with state-of-the-art segmentation and generative networks demonstrate the ability of our networks to capture underlying data distributions in pediatric organs. All the work presented in this thesis can be further

extended, such as training the proposed networks with augmented images. Self-supervision can also be added to our proposed networks to eliminate or reduce the need for annotated organ labels. Additional ablation studies can also be performed to identify useful components of CFG-SegNet.

BIBLIOGRAPHY

[1] D. Ganguly, S. Chakraborty, M. Balitanas, and T.-hoon Kim, “Medical Imaging: A Review,” *SpringerLink*, 15-Sep-2010. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-16444-6_63. [Accessed: 30-Sep-2020].

[2] J. H. Scatliff and P. J. Morris, “From Roentgen to magnetic resonance imaging: the history of medical imaging,” *North Carolina medical journal*. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/24663131/. [Accessed: 30-Sep-2020].

[3] M. Pearce et al., "Radiation exposure from CT scans in childhood and subsequent risk of leukaemia and brain tumours: a retrospective cohort study", *The Lancet*, vol. 380, no. 9840, pp. 499-505, 2012. Available: 10.1016/s0140-6736(12)60815-0.

[4] "Radiography | FDA", *U.S. Food and Drug Administration*, 2020. [Online]. Available: https://www.fda.gov/radiation-emitting-products/medical-x-ray-imaging/radiography. [Accessed: 30- Sep- 2020].

[5] P. Lukashevich, B. Zalesky and S. Ablameyko, "Medical image registration based on SURF detector", *Pattern Recognition and Image Analysis*, vol. 21, no. 3, pp. 519-521, 2011. Available: 10.1134/s1054661811020696.

[6] M. Arbib, *Handbook of Brain Theory and Neural Networks*. Cambridge, MA: The MIT Press, 2016, pp. 255-258.

[7] A. Ismail, T. Rahmat and S. Aliman, "Chest X-Ray Image Classification Using Faster R-CNN”, *Malaysian Journal of Computing*, vol. 4, no. 1, pp. 225-236, 2019.

[8] A. Ismail, T. Rahmat and S. Aliman, "CHEST X-RAY IMAGE CLASSIFICATION USING FASTER R-CNN", *Semanticscholar.org*, 2020. [Online]. Available: https://www.semanticscholar.org/paper/CHEST-X-RAY-IMAGE-CLASSIFICATION-USING-FASTER-R-CNN-Ismail-Rahmat/cb02b8fcddeac6f9932ab024e56b7572b90a584e. [Accessed: 30- Sep- 2020].

[9] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.

[10] O. Ronneberger, P. Fischer, T. Brox: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing (2015)

[11] M. H. Hesamian, W. Jia, X. He, and P. Kennedy, “Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges,” *J Digit Imaging*, vol. 32, no. 4, pp. 582–596, Aug. 2019, doi: 10.1007/s10278-019-00227-x.

[12] E. Gibson *et al.*, “Automatic Multi-Organ Segmentation on Abdominal CT With Dense V-Networks,” *IEEE Trans. Med. Imaging*, vol. 37, no. 8, pp. 1822–1834, Aug. 2018, doi: 10.1109/TMI.2018.2806309.

[13] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, “Deep learning on image denoising: An overview,” *Neural Networks*, vol. 131, pp. 251–275, 2020, doi: https://doi.org/10.1016/j.neunet.2020.07.025.

[14] L. Gondara, “Medical image denoising using convolutional denoising autoencoders,” *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 241–246, Dec. 2016, doi: 10.1109/ICDMW.2016.0041.

[15] H. S. Park, J. Baek, S. K. You, J. K. Choi, and J. K. Seo, “Unpaired image denoising using a generative adversarial network in X-ray CT,” *IEEE Access*, vol. 7, pp. 110414–110425, 2019, doi: 10.1109/ACCESS.2019.2934178.

[16] J. S. Isaac and R. Kulkarni, "Super resolution techniques for medical image processing," *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, Mumbai, 2015, pp. 1-6, doi: 10.1109/ICTSD.2015.7095900.

[17] Y. Chen, F. Shi, A. G. Christodoulou, Y. Xie, Z. Zhou, and D. Li, “Efficient and Accurate MRI Super-Resolution Using a Generative Adversarial Network and 3D Multi-level Densely Connected Network,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, vol. 11070, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham: Springer International Publishing, 2018, pp. 91–99.

[18] Y. Zhang and M. An, “Deep Learning- and Transfer Learning-Based Super Resolution Reconstruction from Single Medical Image,” *Journal of Healthcare Engineering*, vol. 2017, pp. 1–20, 2017, doi: 10.1155/2017/5859727.

[19] P. Isola, J. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 5967-5976, doi: 10.1109/CVPR.2017.632.

[20] J. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 2242-2251, doi: 10.1109/ICCV.2017.244.

[21] X. Yi, E. Walia, and P. Babyn, “Generative Adversarial Network in Medical Imaging: A Review,” *Medical Image Analysis*, vol. 58, p. 101552, Dec. 2019, doi: 10.1016/j.media.2019.101552.

[22] P. Costa et al., "End-to-End Adversarial Retinal Image Synthesis," in IEEE Transactions on Medical Imaging, vol. 37, no. 3, pp. 781-791, March 2018, doi: 10.1109/TMI.2017.2759102.

[23] A. Bissoto, F. Perez, E. Valle, and S. Avila, “Skin Lesion Synthesis with Generative Adversarial Networks,” in *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, 2018, pp. 294–302.

[24] Z. Peng *et al.*, “A Method of Rapid Quantification of Patient-Specific Organ Doses for CT Using Deep-Learning based Multi-Organ Segmentation and GPU-accelerated Monte Carlo Dose Computing,” *Medical Physics*, vol. 47, Mar. 2020, doi: 10.1002/mp.14131.

[25] T. G. Schmidt, A. S. Wang, T. Coradi, B. Haas, and J. Star-Lack, “Accuracy of patient-specific organ dose estimates obtained using an automated image segmentation algorithm,” *Journal of Medical Imaging*, vol. 3, p. 9, 2016.

[26] P. Jackson, N. Hardcastle, N. Dawe, T. Kron, M. S. Hofman, and R. J. Hicks, “Deep Learning Renal Segmentation for Fully Automated Radiation Dose Estimation in Unsealed Source Therapy,” *Front. Oncol.*, vol. 8, p. 215, Jun. 2018, doi: 10.3389/fonc.2018.00215.

[27] V. Ferrari *et al.*, “Value of multidetector computed tomography image segmentation for preoperative planning in general surgery,” *Surg Endosc*, vol. 26, no. 3, pp. 616–626, Mar. 2012, doi: 10.1007/s00464-011-1920-x.

[28] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[29] S. Vesal, N. Ravikumar, and A. Maier, “A 2D dilated residual U-Net for multi-organ segmentation in thoracic CT,” *arXiv:1905.07710 [cs, eess]*, May 2019, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/1905.07710.

[30] Y. Wang, Y. Zhou, W. Shen, S. Park, E. K. Fishman, and A. L. Yuille, “Abdominal multi-organ segmentation with organ-attention networks and statistical fusion,” *Medical Image Analysis*, vol. 55, pp. 88–102, Jul. 2019, doi: 10.1016/j.media.2019.04.005.

[31] Z. Gu *et al.*, “CE-Net: Context Encoder Network for 2D Medical Image Segmentation,” *IEEE Trans. Med. Imaging*, vol. 38, no. 10, pp. 2281–2292, Oct. 2019, doi: 10.1109/TMI.2019.2903562.

[32] T. Okada *et al.*, “Multi-organ segmentation in abdominal CT images,” in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, San Diego, CA, Aug. 2012, pp. 3986–3989, doi: 10.1109/EMBC.2012.6346840.

[33] Z. Xu *et al.*, “Efficient multi-atlas abdominal segmentation on clinically acquired CT with SIMPLE context learning,” *Medical Image Analysis*, vol. 24, no. 1, pp. 18–27, Aug. 2015, doi: 10.1016/j.media.2015.05.009.

[34] F. Milletari, N. Navab and S. Ahmadi, "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation," *2016 Fourth International Conference on 3D Vision (3DV)*, Stanford, CA, 2016, pp. 565-571, doi: 10.1109/3DV.2016.79.

[35] X. Zhou, T. Ito, R. Takayama, S. Wang, T. Hara, and H. Fujita, “Three-Dimensional CT Image Segmentation by Combining 2D Fully Convolutional Network with 3D Majority Voting,” in *Deep Learning and Data Labeling for Medical Applications*, vol. 10008, G. Carneiro, D. Mateus, L. Peter, A. Bradley, J. M. R. S. Tavares, V. Belagiannis, J. P. Papa, J. C. Nascimento, M. Loog, Z. Lu, J. S. Cardoso, and J. Cornebise, Eds. Cham: Springer International Publishing, 2016, pp. 111–120.

[36] C. Rao and Y. Liu, “Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization,” *Computational Materials Science*, vol. 184, p. 109850, 2020, doi: 10.1016/j.commatsci.2020.109850.

[37] M. Larsson, Y. Zhang, and F. Kahl, “Robust Abdominal Organ Segmentation Using Regional Convolutional Neural Networks,” 2017, pp. 41–52, doi: 10.1007/978-3-319-59129-2_4.

[38] H. R. Roth *et al.*, “Hierarchical 3D fully convolutional networks for multi-organ segmentation,” *arXiv:1704.06382 [cs]*, Apr. 2017, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/1704.06382.

[39] M. Shahedi *et al.*, “Segmentation of uterus and placenta in MR images using a fully convolutional neural network,” in *Medical Imaging 2020: Computer-Aided Diagnosis*, Houston, United States, Mar. 2020, p. 59, doi: 10.1117/12.2549873.

[40] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng, “VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images,” *NeuroImage*, vol. 170, pp. 446–455, 2018, doi: https://doi.org/10.1016/j.neuroimage.2017.04.041.

[41] H. Kim *et al.*, “Abdominal multi-organ auto-segmentation using 3D-patch-based deep convolutional neural network,” *Sci Rep*, vol. 10, no. 1, p. 6204, Dec. 2020, doi: 10.1038/s41598-020-63285-0.

[42] D. Karimi, G. Samei, Y. Shao, and S. Salcudean, “A deep learning-based method for prostate segmentation in T2-weighted magnetic resonance imaging,” *arXiv:1901.09462 [cs, eess, stat]*, Dec. 2019, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/1901.09462.

[43] I. J. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Cambridge, MA, USA, 2014, pp. 2672–2680.

[44] M. Mirza and S. Osindero, “Conditional Generative Adversarial Nets,” *arXiv:1411.1784 [cs, stat]*, Nov. 2014, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/1411.1784.

[45] A. Odena, C. Olah, and J. Shlens, “Conditional Image Synthesis with Auxiliary Classifier GANs,” International Convention Centre, Sydney, Australia, Aug. 2017, vol. 70, pp. 2642–2651, [Online]. Available: http://proceedings.mlr.press/v70/odena17a.html.

[46] H. Zhao, O. Gallo, I. Frosio and J. Kautz, "Loss Functions for Image Restoration With Neural Networks," in IEEE Transactions on Computational Imaging, vol. 3, no. 1, pp. 47-57, March 2017, doi: 10.1109/TCI.2016.2644865.

[47] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” 2018, [Online]. Available: https://openreview.net/forum?id=Hk99zCeAb.

[48] H. Tang, D. Xu, W. Wang, Y. Yan, and N. Sebe, “Dual Generator Generative Adversarial Networks for Multi-domain Image-to-Image Translation,” in *Computer Vision – ACCV 2018*, 2019, pp. 3–21.

[49] A. Radford, L. Metz, and S. Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,” *CoRR*, vol. abs/1511.06434, 2016.

[50] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “ClusterGAN: Latent Space Clustering in Generative Adversarial Networks,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4610–4617, 2019, doi: 10.1609/aaai.v33i01.33014610.

[51] Z. Yi, H. Zhang, P. Tan, and M. Gong, “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Oct. 2017, pp. 2868–2876, doi: 10.1109/ICCV.2017.310.

[52] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein Generative Adversarial Networks,” International Convention Centre, Sydney, Australia, Aug. 2017, vol. 70, pp. 214–223, [Online]. Available: http://proceedings.mlr.press/v70/arjovsky17a.html.

[53] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, “On Convergence and Stability of GANs,” *arXiv:1705.07215 [cs]*, Dec. 2017, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/1705.07215.

[54] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim and J. Choo, "StarGAN: Unified Generative Adversarial Networks for Multi-domain Image-to-Image Translation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 8789-8797, doi: 10.1109/CVPR.2018.00916.

[55] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks,” International Convention Centre, Sydney, Australia, Aug. 2017, vol. 70, pp. 1857–1865, [Online]. Available: http://proceedings.mlr.press/v70/kim17a.html.

[56] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved Techniques for Training GANs,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2016, pp. 2234–2242.

[57] S. Jenni and P. Favaro, “On Stabilizing Generative Adversarial Training With Noise,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 12137–12145, doi: 10.1109/CVPR.2019.01242.

[58] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, “Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2016, pp. 82–90.

[59] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, [Online]. Available: http://arxiv.org/abs/1412.6980.

[60] J. L. Lerwick, “Minimizing pediatric healthcare-induced anxiety and trauma,” *WJCP*, vol. 5, no. 2, p. 143, 2016, doi: 10.5409/wjcp.v5.i2.143.

[61] “Radiation Risks and Pediatric Computed Tomography,” *National Cancer Institute*. [Online]. Available: https://www.cancer.gov/about-cancer/causes-prevention/risk/radiation/pediatric-ct-scans. [Accessed: 30-Sep-2020].

[62] B. B. Thukral, “Problems and preferences in pediatric imaging,” *Indian J Radiol Imaging*, vol. 25, no. 4, pp. 359–364, 2015, doi: 10.4103/0971-3026.169466.

[63] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” Long Beach, California, USA, Jun. 2019, vol. 97, pp. 6105–6114, [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html.

[64] M. Chetoui and M. A. Akhloufi, "Explainable Diabetic Retinopathy using EfficientNET*," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 2020, pp. 1966-1969, doi: 10.1109/EMBC44109.2020.9175664.

[65] F. Kanavati *et al.*, “Weakly-supervised learning for lung carcinoma classification using deep learning,” *Sci Rep*, vol. 10, no. 1, p. 9297, Dec. 2020, doi: 10.1038/s41598-020-66333-x.

[66] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” Lille, France, Jul. 2015, vol. 37, pp. 448–456, [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html.

[67] J. Frankle, D. J. Schwab, and A. S. Morcos, “Training BatchNorm and Only BatchNorm: On the Expressive Power of Random Features in CNNs,” *arXiv:2003.00152 [cs, stat]*, Jun. 2020, Accessed: Sep. 30, 2020. [Online]. Available: http://arxiv.org/abs/2003.00152.

[68] J. D. Curtó, I. C. Zarza, F. D. la Torre, I. King, and M. R. Lyu, “High-Resolution Deep Convolutional Generative Adversarial Networks,” *CoRR*, vol. abs/1711.06491, 2017, [Online]. Available: http://arxiv.org/abs/1711.06491.

[69] J. P. Turner and T. Nowotny, “Estimating numerical error in neural network simulations on Graphics Processing Units,” *BMC Neurosci*, vol. 16, no. S1, pp. P182, 1471-2202-16-S1-P182, Dec. 2015, doi: 10.1186/1471-2202-16-S1-P182.

[70] C. N. Enoch Kan, N. Maheenaboobacker and D. H. Ye, "Age-Conditioned Synthesis of Pediatric Computed Tomography with Auxiliary Classifier Generative Adversarial Networks," 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), Iowa City, IA, USA, 2020, pp. 109-112, doi: 10.1109/ISBI45749.2020.9098623.

[72] K. Marstal, F. Berendsen, M. Staring and S. Klein, "SimpleElastix: A User-Friendly, Multi-lingual Library for Medical Image Registration," 2016 IEEE Conference on

Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, 2016, pp. 574-582, doi: 10.1109/CVPRW.2016.78.

[73] “LightSpeed VCT,” *GE Healthcare Systems*. [Online]. Available: https://www.gehealthcare.com/courses/lightspeed-vct. [Accessed: 30-Sep-2020].

[74] “XNAT: Home,” *XNAT*. [Online]. Available: https://www.xnat.org/. [Accessed: 30-Sep-2020].

[75] “Medical Image Processing, Analysis and Visualization,” *Center for Information Technology*. [Online]. Available: https://mipav.cit.nih.gov/. [Accessed: 30-Sep-2020].

[76] *3D Slicer*. [Online]. Available: https://www.slicer.org/. [Accessed: 30-Sep-2020].

[77] *C.8.8 Radiotherapy Modules*. [Online]. Available: http://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.8.8.html. [Accessed: 30-Sep-2020].

[78] Z. Wang, E. P. Simoncelli and A. C. Bovik, "Multiscale structural similarity for image quality assessment," The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Pacific Grove, CA, USA, 2003, pp. 1398-1402 Vol.2, doi: 10.1109/ACSSC.2003.1292216.

[79] O. Oktay *et al.*, “Attention U-Net: Learning Where to Look for the Pancreas,” *ArXiv*, vol. abs/1804.03999, 2018.

APPENDIX A
SUPPLEMENTARY MATERIALS

Validation and Testing details of section 5.4. Patient numbers (de-identified) for each of the 4 folds are provided. Patients used as atlases in the prostate segmentation experiment are also provided.

	Prostate			Uterus	
	Validation	**Testing**	**Atlas**	**Validation**	**Testing**
Fold 1	122, 125, 116, 74, 63	82, 140, 99, 100, 174, 184	86, 136, 100	182, 179, 90	16, 167, 158
Fold 2	172, 177, 171, 88, 163	159, 110, 52, 89, 169	89, 169, 8	80, 91, 142	147, 85, 157
Fold 3	166, 151, 134, 160, 40	104, 105, 92, 134, 141, 136	104, 130, 136	176, 178, 87	16, 124, 98
Fold 4	94, 38, 170, 67, 126	175, 84, 171, 125, 86, 34	175, 171, 34	150, 127, 180	163, 173, 120

Experimental results of sections 5.3 and 5.4. Average IoUs are computed but are not reported in this thesis since it is redundant to the reported DSCs.

(a) Uterus segmentation results

Network	DSC_avg	DSC_std	IOU_avg	IOU_std	Hausdorf	Hausdorff_std	Fold
Ours	0.79	0.398	0.956	0.0844	0.29	0.998	1
Ours	0.576	0.487	0.935	0.112	1.35	2.25	2
Ours	0.663	0.456	0.966	0.077	0.7426	1.839	3
Ours	0.865	0.311	0.97	0.0719	0.455	1.13	4
CE-Net	0.791	0.383	0.958	0.08	0.285	0.907	1
CE-Net	0.615	0.4558	0.927	0.091	1.498	2.034	2
CE-Net	0.659	0.431	0.943	0.0761	1.411	1.89	3
CE-Net	0.758	0.392	0.958	0.0753	1.106	1.849	4
U-Net	0.771	0.402	0.953	0.0851	0.507	1.302	1
U-Net	0.563	0.483	0.909	0.112	1.77	2.47	2
U-Net	0.649	0.454	0.933	0.0863	1.173	2.261	3
U-Net	0.805	0.338	0.957	0.0832	0.9065	1.644	4

Total	DSC_avg	DSC_std	IOU_avg	IOU_std	Hausdorff_avg	Hausdorff_std
Ours	0.724	0.413	0.952	0.0863	0.709	1.554
CE-Net	0.706	0.415	0.943	0.0806	1.08	1.67
U-Net	0.697	0.419	0.938	0.0917	1.09	1.92

(b) Prostate segmentation (with atlas-based localization) results

Network	DSC_avg	DSC_std	IOU_avg	IOU_std	Hausdorf	Hausdorf	Fold
Ours	0.829	0.295	0.963	0.059	2.429	1.124	1
Ours	0.746	0.355	0.951	0.0619	2.916	0.994	2
Ours	0.8275	0.3045	0.978	0.0371	2.094	0.865	3
Ours	0.674	0.459	0.955	0.0619	2.41	0.8788	4
A-U-Net	0.762	0.386	0.962	0.0613	2.82	1.23	1
A-U-Net	0.749	0.359	0.949	0.064	2.870	1.32	2
A-U-Net	0.821	0.336	0.98	0.0365	1.991	0.908	3
A-U-Net	0.683	0.386	0.948	0.0558	2.896	1.155	4
U-Net	0.802	0.314	0.96	0.0605	2.772	1.357	1
U-Net	0.722	0.374	0.947	0.065	3	1.188	2
U-Net	0.823	0.301	0.975	0.0399	2.6	1.02	3
U-Net	0.627	0.411	0.946	0.0585	2.615	1.091	4

Total	DSC_avg	DSC_std	IOU_avg	IOU_std	Hausdorf	Hausdorff_std
Ours	0.769	0.353	0.964	0.0550	2.462	0.965
A-U-Net	0.754	0.367	0.964	0.0544	2.64	1.15
U-Net	0.7435	0.350	0.957	0.0560	2.75	1.16